Applied Soft Computing 147 (2023) 110835

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Stock price forecasting using PSO hypertuned neural nets and ensembling

Akshat Chauhan, Shivaprakash S.J., Sabireen H., Abdul Quadir Md. *, Neelanarayanan Venkataraman

School of Computer Science and Engineering, Vellore Institute of Technology, Chennai 600127, India

ARTICLE INFO

Article history:
Received 1 May 2023
Received in revised form 30 August 2023
Accepted 6 September 2023
Available online 11 September 2023

Keywords:
Stock price movement
LSTM
GRU
Time series
Artificial intelligence
Nifty 50
Binary classification
Particle swarm optimization

ABSTRACT

The stock market is a platform that allows individuals and organizations to buy stocks of publicly listed companies. It is imperative for investors and traders to utilize the platform to buy and sell stocks efficiently, but they must also determine when to do it in order to maximize profits. As trading involves holding stocks for shorter periods, projecting the future direction of a stock's price becomes essential. In recent years, deep neural network-based trading strategies have been researched and implemented to identify when a stock's price will increase or decrease. The main issues in implementing such solutions are that they need to deal with the noisy nature of the stock market and the problem of overfitting. The objective of the paper is to put forth an approach that utilizes deep learning techniques to predict price movements in the Nifty 50 index. The paper will explore the use of Recurrent Neural Networks (RNNs) for the given task. The paper will also look into applying metaheuristic algorithms to further improve the results of the prediction models. In this approach, RNNs, including Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU), are utilized to predict the movement of the index. The models are trained on a unique and efficient feature set that takes into consideration the stock price of large market capitalization companies present in the National Stock Exchange (NSE). Our findings show that ensembled architectures produce better results than individual models. An LSTM and GRU ensembled architecture produced an accuracy of 56.66% and a precision of 0.4734. Particle swarm optimization (PSO) was put forth as a method to hypertune the models to improve their performance. The LSTM, hypertuned with PSO, produced an accuracy of 57.64% and a precision of 0.2882. To further enhance the model's stock price prediction performance, the LSTM and GRU ensembled architecture was ensembled with the PSO hypertuned LSTM architecture to produce a model that gives the highest accuracy of 57.72%. The proposed ensemble approach outperforms the other cutting edge techniques used to forecast how the stock price of the NSE will move. Additionally, the ensemble method increased precision from 0.2882 to 0.5485, demonstrating that ensembling and the PSO algorithm combine to produce models with superior performance. Based on the results, combining PSO hyper parameter optimized models with ensembling provides a good approach towards price movement predictions and also shows the potential of using this approach in other Artificial Intelligence (AI) fields to improve the performance of deep learning models.

Code metadata

Permanent link to reproducible Capsule: https://doi.org/10.24433/CO.7235588.v1.

The code (and data) in this article has been certified as Reproducible by Code Ocean: (https://codeocean.com/). More information on the Reproducibility Badge Initiative is available at https://www.elsevier.com/physical-sciences-and-engineering/computer-science/journals.

* Corresponding author.
E-mail addresses: akshat.chauhan2020@vitstudent.ac.in (Akshat Chauhan), shivaprakash.s2020@vitstudent.ac.in (Shivaprakash S.J.), sabireen.h2019@vitstudent.ac.in (Sabireen H.), abdulquadir.md@vit.ac.in (Abdul Quadir Md.), neelanarayanan.v@vit.ac.in (Neelanarayanan V.).

https://doi.org/10.1016/j.asoc.2023.110835

1. Introduction

An individual can purchase equities in publicly traded companies on the stock market. Through this platform, a person can purchase, sell, and trade stocks. Several stock exchanges exist in India, including the National Stock Exchange (NSE) and the Bombay Stock Exchange (BSE). Both of the equity exchanges have the same functionality, with the difference being that the BSE has 5246 listed stocks as of 8th February 2022 and the NSE has 1817 listed stocks as of 8th July 2022. Both investors and traders use the exchange to make profits. Investing involves buying assets and holding on to them with the expectation that over time the held asset's value will rise, resulting in the individual getting good returns on the invested amount. Investment is primarily done in a few categories, which include stocks, bonds, commodities, mutual funds, etc. Investors buy stocks, take ownership of the stock, and make profit by selling them at a higher price. Investing is often done for the long term as the benefits of compounding are only achieved with time. Trading also involves buying and selling assets to make a profit. Trading can be divided into two types, which are: going short and going long. When a trader goes long, the trader intends to earn profits by acquiring assets for a lower price and selling them for a higher price. When a trader shorts an asset, the trader is involved in selling an asset at a higher price and buying it at a lower price. This is achieved by essentially borrowing the asset and selling it and buying the asset in the future Traders take a long position when they expect the asset to increase in value in the future. When a trader expects that an asset will have a decrease in value in the future, the trader takes a short position. Traders require knowledge on stock price movement to make accurate decisions that generate profit.

Stock price movement prediction refers to predicting if a stock will have an increase or decrease in value in the future. A stock's future price exceeding its current price is referred to as a positive price movement. A stock's future price being less than its current price is referred to as a negative price movement. Traditional trading strategies to predict stock price movement involve using technical indicators, which are patterns found by mathematical calculations. Technical indicators are used to make decisions on buying and selling a stock. These indicators are made by using the historical data of the stock. Some of the popular technical indicators used in trading are moving average convergence divergence (MACD) [1], Bollinger Bands [2], candlestick patterns [3], simple moving average indicator (SMA), and exponential moving average indicator (EMA). Modern trading strategies involve machine learning based approaches to predict stock price movement. A supervised learning approach using techniques like Artificial Neural Networks (ANNs) [4], support vector machines (SVMs) [5], random forest, deep learning, and ensemble techniques [6] are employed to find underlying patterns in the provided data. The discovered patterns can be used to classify the model based on whether the predicted movement is positive or negative. Some of the input features include sentiment analysis of news reports, tweets, etc., technical indicators, and historical data of the stock.

In the realm of artificial intelligence (AI), Particle Swarm Optimization (PSO) is a popular optimization technique that can be used to address challenging issues. Stock price prediction is one of the popular uses of PSO in AI, where it is applied to optimize the settings of various machine learning algorithms. The successful application of PSO in stock price prediction has shown that it is effective in enhancing the precision and effectiveness of AI systems. PSO can also be used with a variety of other AI techniques, such as anomaly detection, natural language processing, and computer vision. Researchers can improve the functionality of these systems and produce more accurate and trustworthy findings by introducing PSO into AI models. Machine learning based approaches have great potential in predicting the movement of stocks. A great advantage of using a machine learning based approach is that it can detect hidden patterns in data. The movement of stock prices may be additionally predicted using these hidden patterns. Some of the problems faced in current research includes: achieving good accuracy in forecasting stock price movement. It is difficult to forecast the movement of a given stock due to its noisy and non-stationary nature. Selecting the right set of features to feed to a machine learning model and ensuring computational efficiency. The key contributions of the paper aim to address the problems mentioned. The following are the main contributions:

- The primary novelty of our work is that we present the first attempt to ensemble a PSO hypertuned model with other advanced architectures like LSTMs and GRUs in order to forecast stock price movements on the NSE. Using a metaheuristic algorithm like PSO improves the time complexity required to hypertune the models, which then helps improve the model's performance.
- This paper introduces a novel set of features that achieve high accuracy while maintaining computational efficiency. These features include the utilization of 20, 50, and 200 day EMAs, along with the stock prices of five major companies in the NSE and stock price details related to the NSE.
- In an effort to accelerate the training of our models, the proposed method focused on a limited set of features instead of examining numerous technical indicators used in stock analysis.
- Additionally, particle swarm optimization is proposed for fine-tuning RRNs that predict stock price movement with highest accuracy compared to other cutting-edge techniques.
- The results suggest that combing PSO hyper parameter optimized models along with ensembeling shows that there is potential in this approach to enhancing models that can be applied in the other fields of Artificial Intelligence (AI).

The rest of the paper is divided as follows: Section 2 examines the related works to identify current methods for predicting stock price movement, Section 3 defines the problem statement, and Section 4 discusses the proposed methodology. Section 5, presents the experimental setup and results. Section 6 concludes the paper with the future directions.

2. Related work

This section examines the related works with the aim of understanding how stock price movements are identified today. It is further divided into six subsections. The first subsection examines works that used LSTM based architectures. The second subsection examines the work of researchers who used an SVM based approach to stock price movement prediction. The third subsection looks at an ensembling approach. The fourth subsection examines works that used other neural network based architectures. Subsection five examines other machine learning based techniques used by researchers, and finally, the use of batch normalization to enhance model performance is examined in the final subsection.

2.1. Predicting stock price movement using LSTM based architectures

Nelson et al. [4] proposed using LSTM neural networks to forecast stock price movement 15 min into the future. Stock price history, along with technical indicators were supplied to the network as inputs. The proposed model outperformed the random forest architecture but not significantly. This tells us that LSTM on its own might not be good enough for predicting stock price movement. Further research can be done into identifying other algorithms or an ensemble approach to improve performance. Wei et al. [7] suggested using an LSTM Relational Graph Convolution Network (LSTM-RGCN) to predict stock price movement. Overnight stock movement is predicted with the help of news that is released after the market closes for the day. News of related companies is taken into consideration with the help of a stock correlation graph. The model was trained to predict stock price movement of the TPX500 index. The correlation graph enables the algorithm to forecast stock movement for companies that are not mentioned in the news. This is something that models which look at the news of only the concerned stock are not able to do. This model can only be used for predicting stock movement overnight and might not be useful for traders who want to predict stock movement while the market is open. Nguyen and Seokhoon [8] proposed using transfer learning and an LSTM. The model is first trained on a large corpus of data from multiple stocks. The model is then fine-tuned with the data of the stock of interest. Their research showed that using an input data consisting of stock price of the company of interest, price of the stock market index, and stock price of companies that have similar industrial products as the company of interest provided the best accuracy. This approach is computationally expensive to build due to the use of transfer learning. Further research can be done into combining sentimental features into the training of the model.

Ji et al. [9] proposed using a LSTM model whose performance had been optimized by using Particle Swarm Optimizers (PSO). The dataset used selected is the Australian stock market (ASM). The strength of this work is the proposal of a novel IPSO algorithm that avoids convergence by searching for optimal parameters of the LSTM network. The results indicated that the PSO-LSTM model possessed high reliability and good forecasting capability. Sirignano and Cont [10] created a deep neural network that trained on the feature set of financial markets. The dataset contains information on all orders placed and cancelled for about 1000 NASDAQ stocks. Three LSTM layers make up the neural network. Stochastic Gradient Descent (SGD) is employed to optimize the network. Authors found that feature selection before training reduces the computational complexity. Pang et al. [11] proposed a deep LSTM network to forecast the price movement of a stock. They propose the use of a stock vector, which is created by taking inspiration from word vectors used in natural language processing. The stock vector is used to deal with large amounts of information present for a given stock. The stock vector is utilized to decrease the data's dimensionality and pass it on as inputs to an LSTM. They proposed the use of two LSTM architectures, including one that used an automatic encoder based on stock vectors and one that had an embedding layer based on stock vectors. The models perform well, with the LSTM and embedding layer performing better.

2.2. Predicting stock price movement using SVM based architectures

Ni et al. [5] used SVM to predict stock prices and also fractal feature selection for optimization was carried out. The dataset contained nineteen technical indicators as features. Before processing the data, feature selection was done. They used grid search method to find the best possible combination. They did not consider any macro and micro features and only considered the technical features for training their model. Hao et al. [12] proposed using twin SVM with fuzzy hyperplane to predict stock price movement. Sentiment analysis on articles is done to identify anger words, anxiety words, negative words, etc. These words are used to gain an emotional score of the news article. Latent Dirichlet Allocation (LDA) is applied to understand more abstract topics present in an article. To deal with words that appear infrequently, word2vec is used. To choose features, PSO is employed. The data is then passed to a twin SVM with a fuzzy hyperplane. The model performed well for predicting price movement of stocks in the semiconductor industry. This is quite an impressive model that uses fuzz theory to deal with the influence of noise. Further research can be done to incorporate tweets from Twitter into the prediction process. Kara [13] proposed using ANNs and SVMs for estimating the stock price movement. The Istanbul Stock Exchange was used to make the dataset. The thorough examination of the methods for parameter adjustment is the work's main contribution. The lack of novelty is this work's main shortcoming. The author also did not compare the performance of their model with that of any other model mentioned in previous works. Hasrini et al. [14] proposed using fundamental analysis, technical analysis, sentiment analysis, and historical stock prices to predict stock price movement. While doing research into predicting stock price movement, fundamental analysis is not often taken into consideration. Here, fundamental analysis is done through the use of foreign currency exchange rates. These inputs were passed to an SVM to classify the movement as positive or negative. The stock price movement predictions were done for 9 stocks. The sensitivity analysis showed that for a few stocks, there can be a large standard deviation in accuracy. This shows us that the model is sensitive to the initial data used in training for a few of the stocks. More research can be done to reduce the standard deviation in performance.

2.3. Predicting stock price movement based on ensembling

Li et al. [6] proposed using two recurrent neural networks where one is an LSTM and the other is a GRU. These two models are then connected to a densely connected network to produce a blended ensemble model. In addition to historical data, a sentimental score was given to news titles by using sentiment analysis and was passed on to the blended ensemble model. The proposed model performs well but still has to deal with the volatility in the stock market. The model only used sentiment analysis in addition to historical data of stock prices as input. Further research can be conducted into using other features like the stock price of competitors and other such features. Tsai [15] proposed using ANN (Artificial Neural Network) and DT (Decision Tree) to forecast stock prices in the electron industry in Taiwan. The dataset came from the TEJ database. This paper proposed using ANN with Delta-Rules method. Delta-Rules method means when the ANN learns a new data the weight average in the network will update every time. CART (classification and regression tree) technique was used to create the decision tree model and was used to predict the stock movement. The author trained the 2 models individually and later ensembled them. The ensembled model showed better results than the individual models.

2.4. Predicting stock price movement using neural networks

Lei [16] proposed the use of Wavelet Neural Network (WNN) to predict the trend of a stock. He used Rough set to reduce the number of attributes for optimization. Rough set also played a role in determining the Wavelet Neural Network's structure. Well-known stock market indices make up the dataset for this study. Long et al. [17] suggested using deep learning to predict the stock price movement. The Chinese stock market index was used to train the model proposed by Long. They proposed a multi-filter neural network that uses SGD and back propagation to learn. The author put forth a brand-new model, a hybrid model made up of different types of neural networks. Bhardwaj [18] proposed the use of a convolutional neural network (CNN) to forecast the movement of stock price. The proposed CNN used 1 dimensional convolution layers. The model was trained on the Nifty 50 index. To prevent overfitting, Bhardwaj proposed the use of dropout layers and early stopping. The research shows the potential of CNNs but did not explore the use of 2D convolutional layers or neural network architectures better suited for time series data.

2.5. Other machine learning based approaches to predicting stock price movement

Alsharef et al. [19] introduced frameworks that can be used to forecast time series data. The frameworks were classified into three categories, which are linear model-based frameworks, deep learning-based frameworks, and automated machine learning (AutoML)-based frameworks. AutoML-based frameworks are meant to solve the issue of manually finding the optimal ML model with the optimal hyperparameters by automatically discovering the optimal model and then optimizing its hyperparameters. Some of the AutoML frameworks given in the paper are AutoKeras, Auto-Sklearn, and AutoGluon. Even though the paper provided AutoML as a potential framework, it did not back the potential of AutoML with experiments and results. This leaves room for empirically evaluating AutoML models. Hassan and Nath [20] used Hidden Markov Model for forecasting stock prices of 4 Airlines. The advantage of this paper is that the proposed technique can be implemented with mediocre knowledge. It lacks generalization as it is limited to work within the airline industry and is trained on a very small dataset.

2.6. Application of batch normalization to improve performance

Sergey and Christian [21] propose the use of batch normalization to overcome the problem of internal covariate shift faced while training a deep neural network. Batch normalization involves the process of normalizing each mini batch used in training the model. The proposed method allows models to be trained at higher learning rates and achieve better accuracy in fewer epochs. Sergey and Christian have not explored the application of batch normalization to RNNs. Further research needs to be done to explore the potential of batch normalization in recurrent neural networks like LSTM and GRU. Cooijmans et al. [22] proposed the use of batch normalization in building recurrent neural networks. They applied normalization from the input to the hidden layer of the RNN and also between hidden layers in the RNN. This is done to solve the problem of internal covariate shift. They suggest that using batch normalization will help optimize the model better. The model's performance was tested on different tasks, like question answering and character level language modelling. The results show that using batch normalization helps models converge faster and generalize better.

Hou et al. [23] explored the use of batch normalization to solve the exploding gradient problem faced by LSTMs. They explored the impact of normalization on the LSTM architecture. They examined normalizing layers, weights, and batch normalization. Training of non-normalized LSTMs and normalized LSTMs was done to evaluate the impact of performance. Their results showed that normalized LSTMs were easier to train than their non normalized counterparts and even showed better results. This research reiterates the need for normalization to speed up convergence and improve model performance. Table 1 summarizes the related works examined in this paper based on the algorithms used and their key contributions.

Table 2 examines the parameters present in each related work with respect to the parameter identified as important in this paper.

In summary, the related works discuss various approaches to predicting stock price movement. The approaches taken by researchers can be primarily categorized into 4 approaches, which include LSTM based approaches, SVM based approaches, neural network based approaches, and machine learning based approaches. Researchers used a variety of feature sets to build classifiers. Some of the feature sets were based on sentiment analysis of news reports related to stocks; technical indicators were also taken into consideration in some works; and researchers even proposed examining the performance of related stocks to estimate the movement of a given stock. Researchers suggested the use of PSO algorithms and batch normalization to enhance the performance of model architectures.

3. Problem statement

A stock market is an exchange where investors and traders can buy and sell shares of companies that are publicly traded. When a user buys a stock, the user buys a percentage stake in the company. Therefore, a stock market is essentially used to buy or sell a percentage stake in a company. A particular investor makes money when the price of a stock they own increases and similarly loses money when the price of the stock decreases. Forecasting the movement of stock prices is essential in growing an investor's capital. Due to the noisy characteristics of the samples, predicting stock price movement becomes a challenging task. The movement of a stock is influenced by many factors, such as psychological factors, behaviour, stock prices of rival firms, the behaviour of international markets, etc. Due to the above factors it becomes quite challenging to estimate the movement of a stock with high accuracy. Some of the strategies used to forecast stock price movement include SVM (Support Vector Machine) and deep learning models such as WNN (Wavelet Neural Network), GA (Genetic Algorithms) [24], and ANN (Artificial Neural Networks). This paper aims to identify features and a suitable algorithm that uses the identified features to accurately predict stock price movement. Eq. (1) mathematically defines our problem.

$$M = \begin{cases} 1 & if\ p_{t+1} > p_t \\ 0 & if\ p_{t+1} < p_t \end{cases} \tag{1}$$

Where M is stock price movement, P is stock price, and t is a given time period.

4. Motivation

LSTM [25] is a type of RNN architecture that can effectively process sequential data by selectively retaining or forgetting information based on its relevance to the current task. LSTM networks have a more complex structure than traditional RNNs [26] and include memory cells with input, output, and forget gates, which allow them to learn long-term dependencies in the input data. Overall, LSTM networks are well-suited for a wide range of tasks that involve sequential data. LSTMs can be trained using different techniques, which includes backpropagation and reinforcement learning. LSTMs and GRUs are suited for sequential data and do not give good results with non-sequential data. LSTM and GRU are powerful deep learning models capable of capturing complex temporal patterns and dependencies in sequential data like stock prices, making them more suitable for handling the intricacies of the stock market compared to traditional mathematical models, which may struggle with non-linearity. LSTM and GRU models are designed to learn from historical data and can retain information from previous time steps. This is crucial for stock prediction because past stock prices and trading volumes can provide valuable insights into future movements. By using a data-driven approach, the PSO + LSTM/GRU ensemble can learn from historical trends and adjust its parameters accordingly. For predicting stock movement, PSO + LSTM and PSO + GRU ensembles are used because of their complementing strengths. We take advantage of PSO's global optimization capabilities to fine-tune the models' hyperparameters and initial circumstances by combining it with LSTM and GRU networks. This partnership improves the forecasting accuracy of LSTM and GRU networks by enabling them to efficiently capture complex temporal patterns in stock market data. By combining predictions from various models optimized via PSO, the ensemble aspect further increases variety by reducing overfitting as well as offering more resilient predictions appropriate for the erratic market dynamics.

These groups successfully handle the difficulties posed by anticipating stock movement. Their aptitude for handling acoustic and non-linear financial data is what sets them apart. The

Table 1
Tabulated summary of related works.

Method	Author(s)	Algorithm	Key Contribution
LSTM based architecture	Nelson et al. [4]	LSTM	Proposed the use of stock price history and technical indicators
	Wei et al. [7]	LSTM Sentiment analysis	Proposed using an LSTM Relational Graph Convolution Network
	Nguyen and Seokhoon [8]	LSTM Transfer Learning	Proposed using a pretrained LSTM network
	Ji et al. [9]	LSTM PSO	Proposed the use of a PSO optimized LSTM
	Sirignano and Cont [10]	LSTM Gradient decent	Used feature selection to reduce complexity
	Pang et al. [11]	ALSTM ELSTM	Proposed the use of a stock vector which is inspired by NLP
SVM based architectures	Ni et al. [5]	SVM Feature selection	Proposed the use of SVM and fractal feature selection
	Hao et al. [12]	SVM Sentiment analysis PSO	Proposed using twin SVM with fuzzy hyperplane. Applied PSO for feature selection
	Kara [13]	SVM ANN	Proposed the use of ANN and SVM
	Hasrini et al. [14]	SVM Fundamental & Technical Analysis Sentiment Analysis	Proposed using fundamental, technical, and sentiment analysis, and historical stock prices along with SVM.
Ensembling based approach	Li et al. [6]	LSTM GRU Sentiment Analysis	Proposed using a blended ensemble of LSTM and GRU.
	Tsai [15]	ANN DT	Proposed using ensemble of ANN+DT
Neural network based approach	Lei [16]	WNN Rough set	Proposed the use of Wavelet Neural Network (WNN)
	Long et al. [17]	Multi filter neural network	Proposed a multi-filter neural network with SGD
	Bhardwaj [18]	CNN	Proposed the use of a CNN with dropout and early stopping
Machine Learning based approach	Alsharef et al. [19]	AutoML	Proposed a framework that automatically identifies the optimal ML model
	Hassan and Nath [20]	HMM	Proposed the use of Hidden Markov Model
Batch normalization	Sergey and Christian [21]	Batch normalization	Proposed the use of batch normalization to overcome internal covariate shift
	Cooijmans et al. [22]	Batch normalization RNN	Proposed the use of batch normalization in building RNN.
	Hou et al. [23]	Batch normalization LSTM	Explored the potential of batch normalization to solve the exploding gradient problem.

addition of PSO optimization improves the ability of LSTM and GRU networks to handle the noisy nature of financial data. These networks are intrinsically capable of capturing non-linear trends and dependencies over a range of time scales. Additionally, the ensembles effectively deal with the problem of data scarcity by learning to make predictions even when some data are missing, which happens frequently in financial time series. The dynamic character of LSTM and GRU networks, strengthened by PSO optimization, which enables them to capture both short-term fluctuations and long-term trends, facilitates their capacity to respond to unexpected market shifts. Combining optimization with sequence modelling inside these ensembles yields a reliable method for forecasting stock movement that takes into account the complex and dynamic nature of financial markets.

LSTMs take more time to train when the dataset size is large. This is due to the fact that it is computationally intensive to learn the LSTM's parameters. Hence optimizers are used alongside the deep learning model to find the most optimal set of hyper-parameters and ultimately reach the global minima. Optimizers such as Stochastic Gradient Descent (SGD), AdaGrad, AdaBoost,

Table 2
Parameters present in related works with respect to important parameters proposed in this paper.

Method	Author(s)	Parameters					
		LSTM	GRU	Ensemble	Technical Indicators	PSO	Feature Engineering
LSTM based architecture	Nelson et al. [4]	✓	×	×	✓	×	✓
	Wei et al. [7]	✓	×	×	×	×	✓
	Nguyen and Seokhoon [8]	✓	×	×	×	×	×
	Ji et al. [9]	✓	×	×	×	✓	✓
	Sirignano and Cont [10]	✓	×	×	×	×	✓
	Pang et al. [11]	✓	×	×	×	×	✓
SVM based architectures	Ni et al. [5]	×	×	×	✓	×	✓
	Hao et al. [12]	×	×	×	×	✓	✓
	Kara [13]	×	×	×	×	×	×
	Hasrini et al. [14]	×	×	×	✓	×	✓
Ensembling based approach	Li et al. [6]	✓	✓	✓	×	×	✓
	Tsai [15]	×	×	✓	✓	×	×
Neural network based approach	Lei [16]	×	×	×	×	×	✓
	Long et al. [17]	×	×	×	×	×	×
	Bhardwaj [18]	×	×	×	×	×	×
Machine Learning based approach	Alsharef et al. [19]	×	×	×	×	×	×
	Hassan and Nath [20]	×	×	×	×	×	×
Batch normalization	Sergey and Christian [21]	×	×	×	×	×	×
	Cooijmans et al. [22]	×	×	×	×	×	×
	Hou et al. [23]	✓	×	×	×	×	×

RMSProp, Adam are used to quickly get the best results. But these optimizers may not always reach the global minima. Hence, this paper proposes using Particle Swarm Optimizer (PSO) to hypertune the LSTM and GRU models to achieve improvements over the traditional architectures that only use optimizers.

5. Proposed architecture

A PSO+LSTM and PSO +GRU model ensemble is used along with feature set to predict stock price movement. LSTM or GRU models, when combined with PSO, show promising benefits. The convergence of LSTM/GRU models is enhanced by PSO's ability for global optimization, which facilitates the search for ideal initialization settings and hyperparameters. By taking into account a variety of alternative solutions, the ensemble, which is made up of varied models optimized by PSO, successfully reduces overfitting and strengthens generalization. Furthermore, the PSO's inherent exploration-exploitation balance makes it easier for LSTM/GRU models to understand complex sequential patterns. In the end, the collaborative synergy offers a more thorough knowledge of model uncertainty by addressing the nuances of intricate linkages within data. To establish the effectiveness of this combined strategy, empirical validation is crucial. The extent of improvement, however, depends on variables including dataset features, tuning subtleties, and the difficulty of the problem at hand. For this research, the NIFTY 50 index was used for the prediction of stock movement. Along with the details about NIFTY 50, the closing prices of five large companies in the National Stock Exchange were considered. The inclusion of the closing prices of these companies is based on our hypothesis that the closing prices of these companies have an impact on the price of the NIFTY 50 index. Exponential moving averages are used in many trading strategies, including this information in our feature set would be useful in predicting stock price movement. The twenty-day, fifty-day, and two-hundred-day exponential moving averages were calculated based on the closing price of NIFTY 50 and included in our feature set.

The Nifty 50 index closing price is included in the dataset. After that, the dataset is cleaned by managing missing values, either by deleting them or by putting the mean of all the values in the empty spots. The Data Generation Algorithm is used to calculate and add the 20-day, 50-day, and 200-day EMAs to the dataset after the cleaned data has been passed through it. A training set makes up 80% of the dataset, while the testing set makes up 20%. The PSO+LSTM and PSO+GRU models are trained using the training set. The performance of the model is then evaluated using the testing set (see Fig. 1).

5.1. Data cleaning

The dataset may contain noisy values or may have some missing values. These factors can have an effect on the performance of the model. These need to be handled before the dataset can be used. The common techniques for handling the missing values is replacing it with the mean of the same attribute or by dropping the row in the dataset if the dataset is large. The data is also standardized to bring all the features into a similar range.

5.2. Data generation algorithm

The NIFTY 50 index is used as input by this unit, which then computes the Exponential Moving Averages (EMA) based on the NIFTY 50's daily closing price and adds it to the dataset. A particular sort of moving average called the EMA gives extra weight to recent data items to emphasize how important they are. Due to EMA giving it greater weight, it responds more strongly to recent price movements than a Simple Moving Average (SMA). While passing the data to Recurrent Neural Network models, the data is converted to sequential data with a 20 day window. It is discussed below how to generate EMA data for stock movement prediction (see Fig. 2).

Algorithm 1: EMA Data Generation Algorithm for Stock Movement Prediction

Input: Nifty 50 closing Index

Output: Exponential Moving Average

Pseudocode:

Let n be the total number of data instances

Let the initial time span be 20 days

Loop t=0 to n-1:

EMA_t=(closing_price$_t$ – EMA_{t-1})*2/(time_span +1) + EMA_{t-1}

Difference$_t$ = closing_price$_t$ – closing_price$_{t-1}$

If(Difference$_t$ >0):

Movement$_t$ =1

Else:

Movement$_t$ =0

Add Movement$_t$, Difference$_t$ and EMA_t to the dataset

Repeat the same for time span of 50 and 200 days

5.3. Long short term memory

For training, the aforementioned data was given to a long short-term memory (LSTM). The issue of vanishing gradients that is present in standard recurrent neural networks is addressed by the LSTM model. LSTMs employ gates to keep in mind only pertinent information when training the model. Using sequential data, LSTM models have achieved significant success and are used to forecast stock price movement. LSTM networks feature a more sophisticated structure that contains numerous interconnected layers termed "memory cells", as opposed to standard RNNs, which have a straightforward feedback loop that enables information to be transmitted from one time step to the next. These memory cells have the ability to store and retrieve information for an extended period, which allows the LSTM network to learn long-term dependencies in the input data. The input, output, and forget gates are the three gates that regulate the information flow in each memory cell in an LSTM network. The output gate governs how much of the memory cell's content should be output to the following time step, while the input gate decides how much of the fresh input should be stored in the memory cell. Whatever information should be removed from the memory cell is decided by the forget gate. The LSTM design can efficiently capture both short-term and long-term dependencies, and it can selectively keep or forget information based on its relevance to the current job, making it well-suited for processing sequential data in general.

5.4. Gated recurrent units

A Gated Recurrent Unit Recurrent Neural Network (GRU) received the aforementioned data as well for training. Because to their similar designs and successful performance, GRU is frequently seen as a variation of the LSTM architecture. Moreover, GRU addresses the issue of vanishing gradients that is present in standard recurrent neural networks. The update gate and reset gate are the two gates that the GRU uses to do this. Due to its two gates as opposed to the LSTM's three, the GRU has fewer parameters. What data is sent to the output is determined in part by these two gates. GRU models will be used to forecast changes in stock prices because they have demonstrated excellent performance with sequential and time series data.

5.5. Particle swarm optimizers

PSO [27] is a metaheuristic optimization method that was motivated by the group behaviour of social swarms, including fish schools or bird flocks. PSO searches a search space for the best solution to a given problem using a population of candidate solutions known as particles. According to its own best position and the best position the swarm has so far, each particle's position in the search space and velocity are changed at each iteration. The particles try to steer clear of uncharted territory in the search space and proceed in the direction of the best place they can find. This prevents the algorithm from getting stuck in local minima and enables it to converge towards the global optimum. The PSO algorithm is often used for optimization problems in which the objective function is difficult to evaluate or has many local optima. It has been applied in various fields, including engineering, finance, and data science, and has been shown to be effective in finding near-optimal solutions for complex problems. The PSO algorithm was chosen among other metaheuristic algorithms due to its efficiency. This is primarily because the algorithm has only a few parameters to tune, like particle velocity, number of particles, and particle position. This is particularly important as hypertuning RNNs require more time to train and evaluate when compared to other models. Having an efficient algorithm like PSO helps reduce the computational costs of hypertuning the models. Algorithm 2: PSO Algorithm for Stock Movement Prediction is discussed below.

Algorithm 2: PSO Algorithm for Stock Movement Prediction

Input: n, No. of particles

Output: Best particle

Pseudocode:

Randomly initialize a swarm population of n particles x_i (i=1,2,….n)

Select values for the hyperparameters w,c1,and c2

```
For j in range(max_iter):
    For i in range(n):
    StockSwarm[i].velocity=w*StockSwarm.velocity+r1*c1*(StockSwarm[i].bestPosStock-StockSwarm[i].position)+r2*c2*(Stock_best_pos_swarm-StockSwarm[i].position)
        if StockSwarm[i].velocity<minx:
            Stockswarm[i].velocity=minxv
        elif StockSwarm[i].velocity>maxxv:
            StockSwarm[i].velocity=maxxv
        StockSwarm[i].position+=StockSwarm[i].velocity
        StockSwarm[i].bestFitness=StockSwarm[i].fitness
        StockSwarm[i].bestPos=StockSwarm[i].position
        if StockSwarm[i].fitness<Stock_best_fitness:
            Stock_best_fitness =StockSwarm[i].fitness
            Stock_best_pos =StockSwarm[i].position
        end-For
end-For
Return best particle of Stock swarm
```

Where, Minxv is the lower bound for velocity. This is a limit on the maximum speed that a particle can travel. It prevents particles from overshooting the optimal solution and getting stuck in

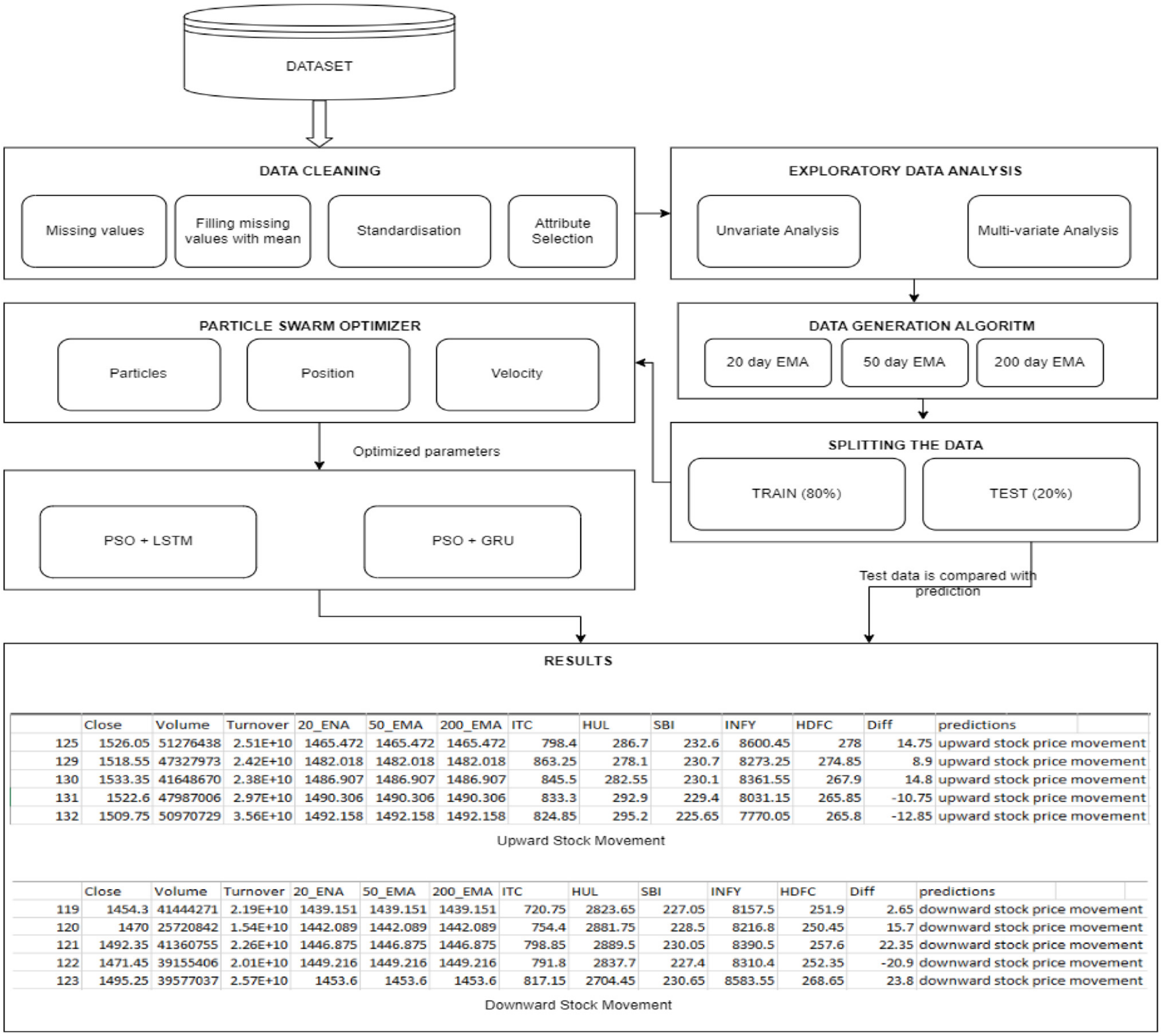


Fig. 1. Block diagram of the proposed model for predicting stock movement.

regions of the search space that are far from the solution. Maxxv is the upper bound for velocity. This is a limit on the minimum speed that a particle can travel. It prevents the particles from taking a long time in reaching the optimal solution and decreases the computation. *N* is the number of particles. This is the number of particles in the swarm, and it determines the size of the search space that is explored. A larger swarm size can help to explore the search space more thoroughly, but it can also increase the computational cost.

w is the inertia. This is a weighting factor that controls the particle's velocity and its tendency to continue in its current direction. It balances the tradeoff between exploration and exploitation in the search space. A high inertia weight value encourages search space exploration, whereas a low one encourages use of the best existing solutions. Understanding of particles is c1. This setting regulates the particle's propensity to migrate towards its ideal position. It establishes the extent to which the particle's experience affects its motion. Whereas a low number stimulates searching the search space, a high value of the cognitive component increases exploitation of the existing position. The social impact of particles is c2. This setting regulates the particle's propensity to migrate towards the world's optimal position. It determines how much each particle's mobility is influenced by the swarm's best solution. Whereas a low number stimulates searching the search space, a high value of the social parameter increases use of the best solution. The maximum number of iterations is expressed as Max iter. The PSO algorithm's flow is shown in Fig. 3.

5.6. PSO and LSTM

The LSTM architecture is hypertuned with the help of the Particle Swarm Architecture. Particle Swarm Optimization is used to identify the optimal number of LSTM units for each hidden layer of the LSTM architecture. A 5 particle PSO was used to identify the optimal number of LSTM units per hidden layer. Eq. (2) to Eq. (5) mathematically describes the above architecture.

$$H = \{h_1 + h_2 \ldots + h_n\} \tag{2}$$

$$L = \tanh(LSTM_H(x)) \tag{3}$$

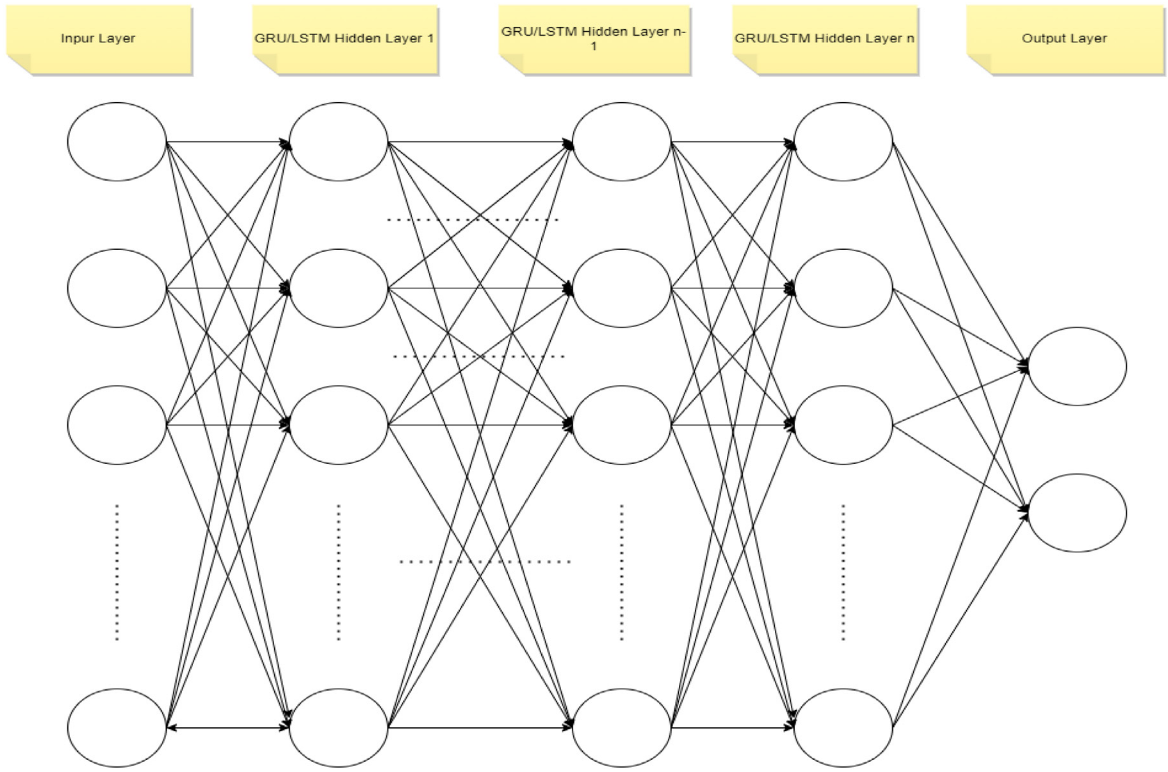


Fig. 2. The LSTM/GRU neural network.

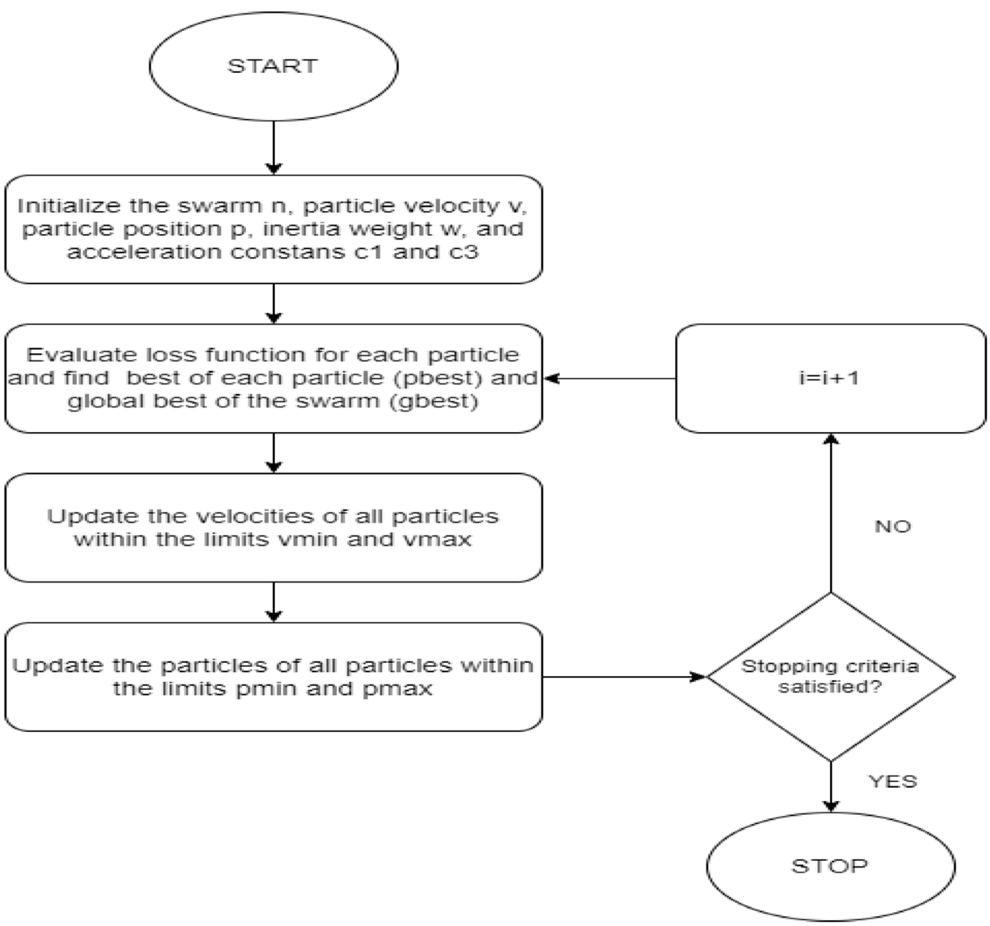


Fig. 3. Flow diagram of PSO algorithm.

$$H_{best} = PSO(H, LSTM, x) \tag{4.1}$$

$$H_{best} = PSO(\{h_1 + h_2 \ldots + h_n\}, \tanh(LSTM_H(x))) \tag{4.2}$$

$$L_{PSO} = \tanh(LSTM_{H_{best}}(X)) \tag{5}$$

Eq. (2) represents the initial hidden units per layer in the LSTM. Eq. (3) is an LSTM that has hidden units H and outputs a prediction for input x. Eq. (4) represents the PSO algorithm, which takes the hidden units, LSTM, and *x* as inputs and returns the optimal hidden units for each layer. Eq. (5) is the PSO hypertuned LSTM, which is made with the optimal number of hidden units.

Time complexity

The PSO algorithm has a time complexity of O(NP*D)
where *N* is the number of particles,
P is the number of iterations,
D is the dimensionality of the problem.

The time complexity of a single LSTM cell is O(D^2), where D is the input dimensionality. The time complexity of an LSTM layer with *N* cells is O(ND^2), and the time complexity of an LSTM network with L layers is O(LN*D^2).

5.7. PSO and GRU

The Particle Swarm Architecture assists in the hypertuning of the GRU architecture as well. At each hidden layer of the GRU architecture, the algorithm aids in determining the ideal number of GRU units. The ideal number of GRU units was determined using a five particle PSO, and the hypertuned model was built using the results. The aforementioned architecture is mathematically described by Eqs. (6) through (9).

$$H = \{h_1 + h_2 \ldots + h_n\} \tag{6}$$

$$G = \tanh(GRU_H(x)) \tag{7}$$

$$H_{best} = PSO(H, GRU, x) \tag{8.1}$$

$$H_{best} = PSO(\{h_1 + h_2 \ldots + h_n\}, \tanh(GRU_H(x))) \tag{8.2}$$

$$G_{PSO} = \tanh(GRU_{H_{best}}(X)) \tag{9}$$

Eq. (6) represents the initial hidden units per layer in the GRU. Eq. (7) is a GRU that has hidden units H and outputs a prediction for input x. Eq. (8) represents the PSO algorithm, which takes the hidden units, GRU, and *x* as inputs and returns the optimal hidden units for each layer. Eq. (9) is the PSO hypertuned GRU, which is made with the optimal number of hidden units.

Time Complexity

The PSO algorithm has a time complexity of O(NP*D)
where *N* is the number of particles,
P is the number of iterations,
D is the dimensionality of the problem.

The time complexity of a GRU algorithm depends on the number of GRU cells and layers used. The time complexity of a single GRU cell is O(D^2), where D is the input dimensionality. The time complexity of a GRU layer with *N* cells is O(ND^2), and the time complexity of a GRU network with L layers is O(LN*D^2).

5.8. PSO and LSTM ensembled with PSO and GRU ensemble

This model architecture involves ensembling two models whose hyperparameters have already been optimized using PSO. The PSO + LSTM model is ensembled with the PSO + GRU model. This is done to increase the metrics observed in the PSO + LSTM model with the help of the PSO + GRU model. Eq. (10) to Eq. (13) given below mathematically describes the model.

$$L = \tanh(LSTM(x)) \tag{10}$$

$$G = \tanh(GRU(x)) \tag{11}$$

$$E_1 = L \cup G(x) \tag{12.1}$$

$$E_1 = avg(\tanh(LSTM(x), \tanh(GRU(x)))) \tag{12.2}$$

$$L_{PSO} = \tanh(LSTM_{PSO}(x)) \tag{13}$$

$$E = L_{PSO} \cup E_1(x) \tag{14.1}$$

$$E = avg(\tanh(LSTM_{PSO}(x))),\ \ avg(\tanh(LSTM(x), \tanh(GRU(x)))) \tag{14.2}$$

L (Eq. (10)) represents the LSTM model, and G (Eq. (11)) represents the GRU model. E_1 (Eq. (12)) is the ensemble of L and G. Eq. (12.1) explains the ensemble on a high level, while Eq. (12.2) shows in detail how the ensemble is performed. L_{PSO} (Eq. (13)) represents the PSO hypertuned LSTM. E (Eq. (14)) is the final ensemble of L_{PSO} and E_1.

A PSO + LSTM and PSO + GRU ensemble is robust due to several factors. Particle Swarm Optimization (PSO) fine-tunes LSTM and GRU model parameters, enhancing adaptability to varying market conditions. LSTM and GRU's capacity to capture complex temporal patterns ensures generalization to unseen data. The ensemble diversifies predictions, reducing overfitting risk. Moreover, it effectively handles non-linearity, temporal dependencies, and high-dimensional stock market data, leading to improved accuracy and stable performance across different market scenarios. The combined strength of PSO optimization and deep learning models results in a robust and reliable ensemble for stock movement prediction (see Fig. 4).

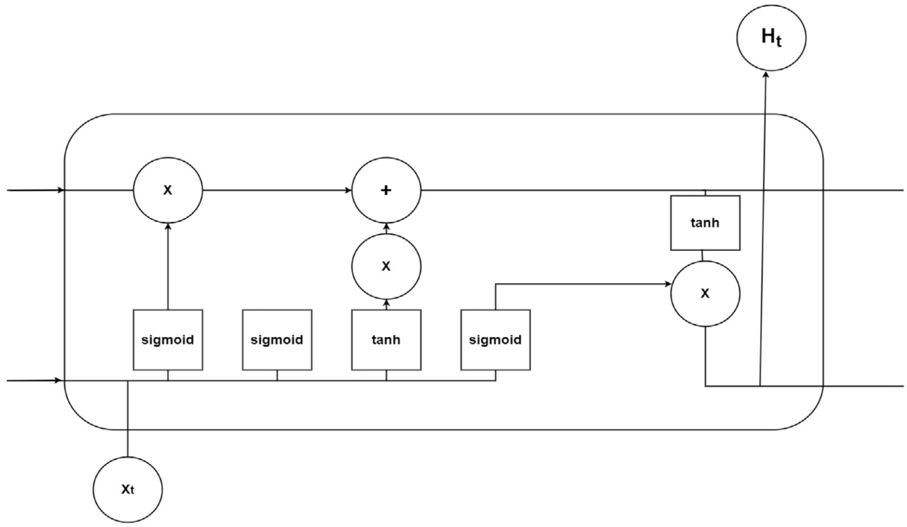
6. Experimental setup and results

6.1. Experimental setup

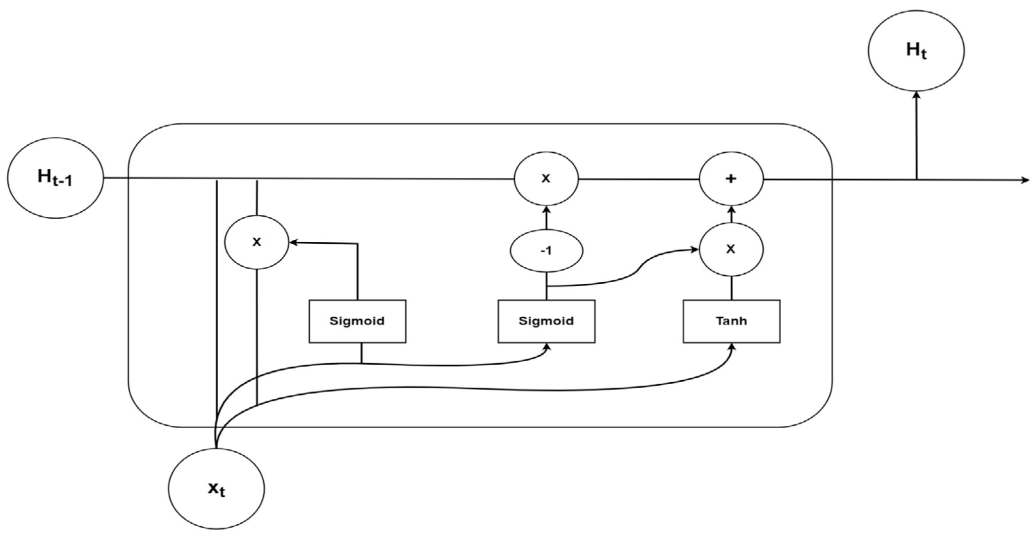
This segment discusses the models used to test our proposed methodology and the hardware required to train the models. The models were trained on an M1 GPU 3.2 GHZ system with 8 GB RAM. The models were defined using Google's TensorFlow. In order to conduct our experiments, data was collected from Yahoo Finance. The data extraction process included collecting daily stock price data of the NSE for the time period of 04/01/2000 to 29/04/2021. Similarly data was collected for five large companies in the NSE like ITC, Hindustan Unilever, and Infosys. The collected data was combined, preprocessed, and used to build the models given in the experiments below.

6.2. Evaluation metrics

Predicting stock price movement is a classification problem where the effectiveness of a model is assessed using common criteria. To assess the model's performance, accuracy, precision, recall, and F1-score are taken into consideration. These measures make use of false positives, false negatives, true positives, and



(a)



(b)

Fig. 4. (a) and (b): The LSTM and GRU unit.

true negatives (fn). The metrics are defined mathematically in Eqs. (15) through (18):

$$\text{Accuracy} = \frac{tp + tn}{tp + fp + tn + fn} \tag{15}$$

$$\text{Precision} = \frac{tp}{tp + fp} \tag{16}$$

$$\text{Recall} = \frac{tp}{tp + fn} \tag{17}$$

$$\text{F1-score} = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{18}$$

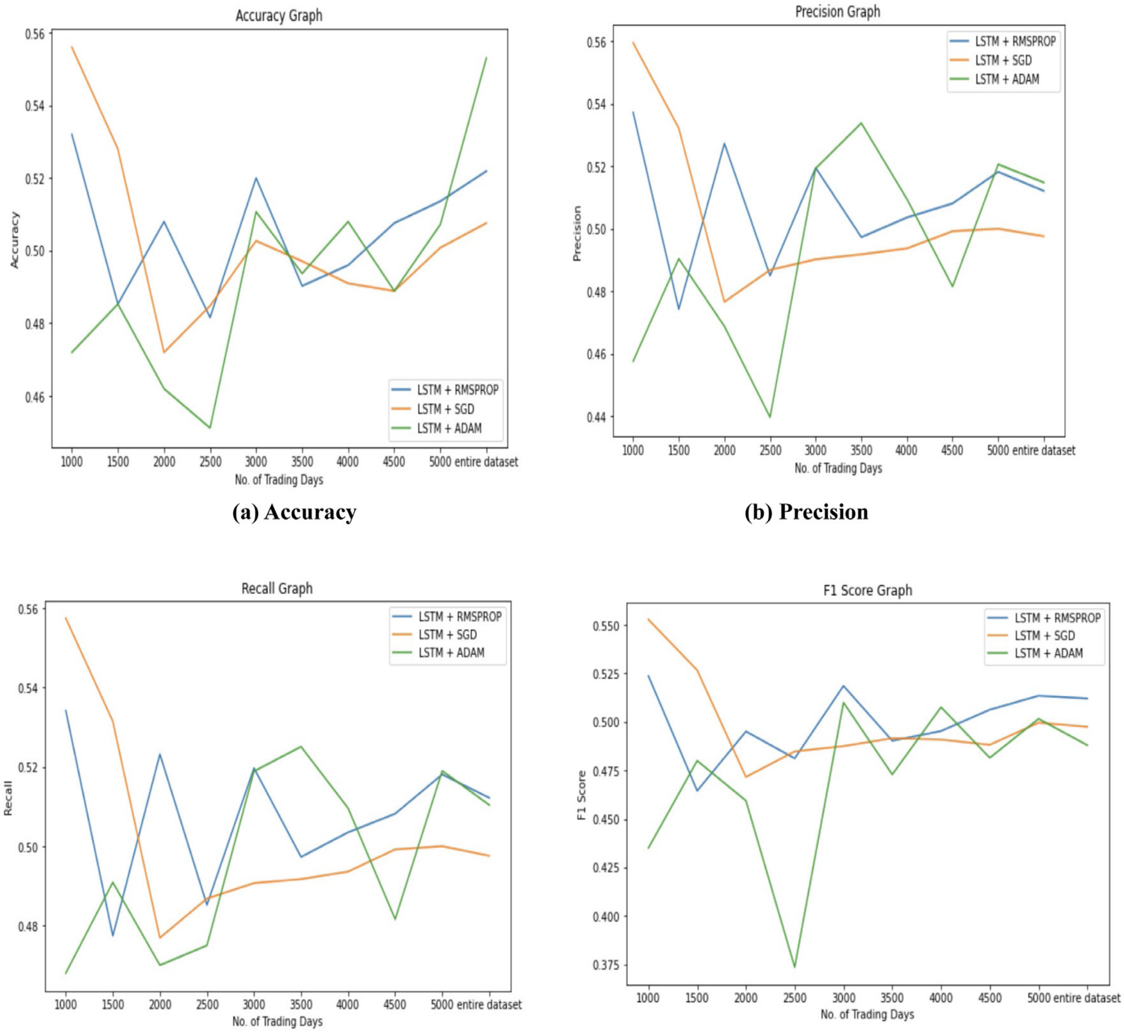
6.3. Experiments on different LSTM and GRU architectures

This section presents the various LSTM and GRU architectures used for stock price prediction in detail. Based on the results of an experiment, more experiments are conducted to identify which model architecture performs the best on the given dataset. The entire dataset consists of 5304 days of trading. For each model, its performance after being trained on a different number of trading days is evaluated and recorded. This helps in exploring and understanding how the model's performance changes based on the number of training days used to train it.

6.3.1. LSTM and optimizer

This experiment trained LSTM architectures with different optimizers to identify the optimizer that has the most promising results. The Adam optimizer, SGD optimizer, and RMSprop optimizer are taken into consideration in this experiment. Each LSTM model is trained on stock data for a given number of trading days to identify how the model performs with different sizes of data. Each model is trained for 50 epochs and then evaluated on a test dataset. Table 3 to Table 5 and Fig. 5 contain the experiment's results.

From observing Tables 3–5 and Fig. 5, it is observed that LSTM + Adam initially did not perform as well as the other optimizers, but with the increase in the size of the dataset, Adam has performed better than the other optimizers. Both RMSprop and SGD show a decline in performance with an increase in dataset size. Based on these results, the next set of experiments will use Adam as the chosen optimizer.



(a) Accuracy **(b) Precision**

(c) Recall **(d) F1 Score**

Fig. 5. 5(a)–5(d): Performance of LSTM and Optimizers.

Table 3
LSTM and Adam model architecture performance.

No. of Trading Days	Accuracy	Precision	Recall	F1 Score
1000	0.472	0.4576	0.468	0.435
1500	0.4853	0.4904	0.4909	0.48
2000	0.462	0.4688	0.47	0.4594
2500	0.4512	0.4397	0.475	0.3736
3000	0.5107	0.5193	0.5189	0.51
3500	0.4937	0.5338	0.5251	0.4729
4000	0.508	0.5095	0.5096	0.5076
4500	0.4889	0.4815	0.4816	0.4815
5000	0.5072	0.5206	0.519	0.5016
entire dataset	0.553	0.5148	0.5104	0.488

Table 4
LSTM and SGD model architecture performance.

No. of Trading Days	Accuracy	Precision	Recall	F1 Score
1000	0.556	0.5595	0.5575	0.5528
1500	0.528	0.5322	0.5315	0.5267
2000	0.472	0.4766	0.4769	0.4716
2500	0.4848	0.4868	0.4868	0.4848
3000	0.5027	0.4902	0.4907	0.4875
3500	0.4971	0.4918	0.4917	0.4916
4000	0.491	0.4937	0.4936	0.4909
4500	0.4889	0.4992	0.4992	0.4882
5000	0.5008	0.5	0.5	0.4996
entire dataset	0.5076	0.4976	0.4976	0.4975

6.3.2. LSTM with adam and regularization

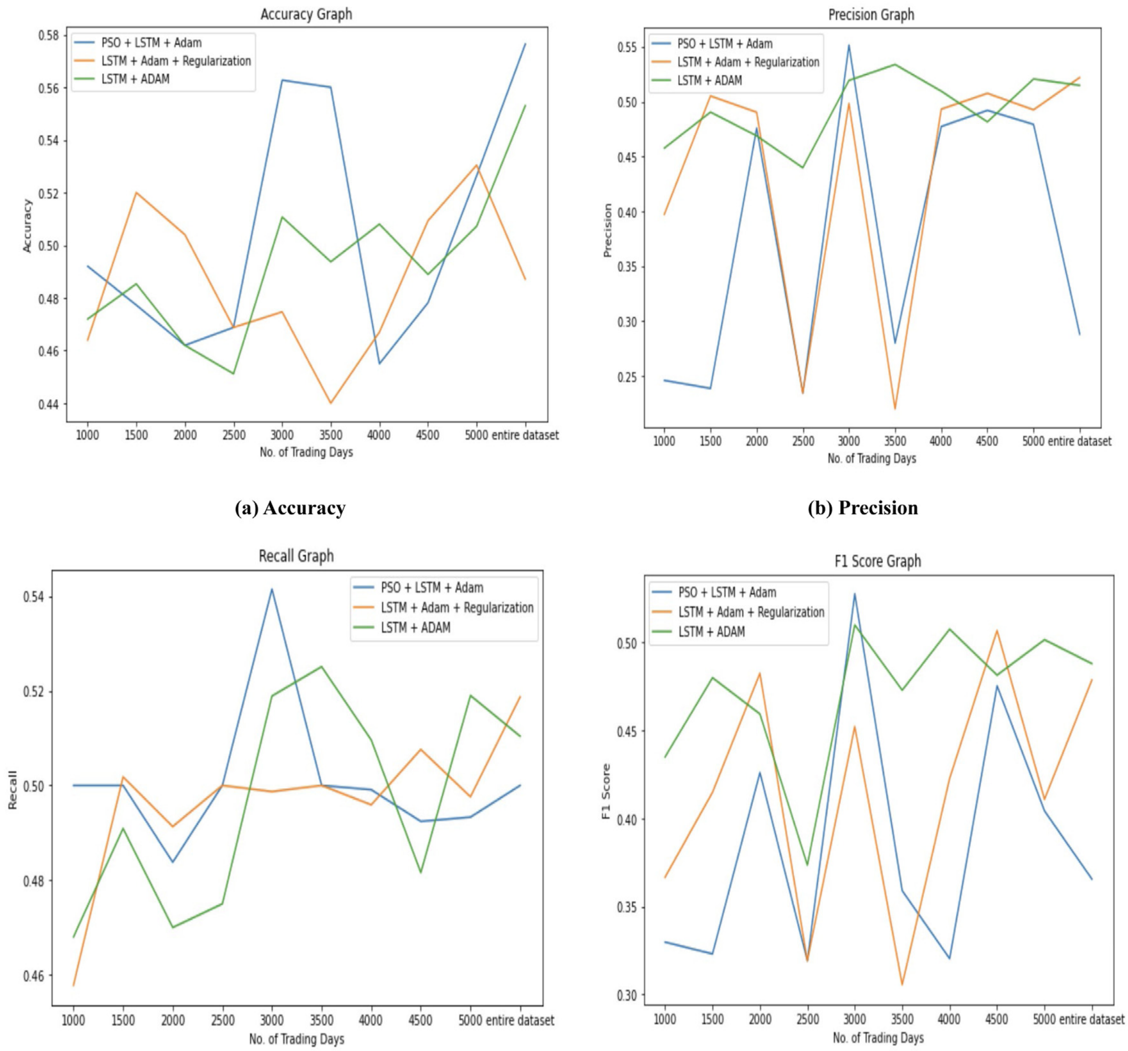
The previous LSTM architectures did not make use of regularization. The current architecture includes the use of dropout and batch normalization layers in the model architecture. This is done to help the model improve its generalization capability. The results of the experiment are given in Table 6.

The results of Table 6 show that regularizing the LSTM model was not enough to improve the performance of the model. The regularized model showed good performance initially but showed a decline in performance with the increase in dataset size. To further improve performance, the number of units in each hidden layer needs to be hypertuned.

6.3.3. PSO with LSTM and Adam

PSO is used to identify the optimal number of LSTM units for each hidden layer of the LSTM architecture. A 5 particle PSO was used to identify the optimal number of LSTM units per hidden layer. The results of the hypertuned models are given in Table 7.

The data in Tables 6–7 and Fig. 6, which describe the performance of the architectures, shows that the LSTM architecture that was hypertuned with PSO gives better accuracy than the other



(a) Accuracy

(b) Precision

(c) Recall

(d) F1 Score

Fig. 6. 6(a)–6(d): Performance of PSO with LSTM and Adam.

Table 5
LSTM and RMSprop model architecture performance.

No. of Trading Days	Accuracy	Precision	Recall	F1 Score
1000	0.532	0.5372	0.5342	0.5237
1500	0.4853	0.4743	0.4774	0.4645
2000	0.508	0.5273	0.5232	0.4951
2500	0.4816	0.485	0.4852	0.4812
3000	0.52	0.5195	0.5197	0.5186
3500	0.4903	0.4973	0.4973	0.4903
4000	0.496	0.5036	0.5035	0.4953
4500	0.5076	0.5081	0.5082	0.5063
5000	0.5136	0.5182	0.5181	0.5135
entire dataset	0.5219	0.5121	0.5122	0.5121

Table 6
LSTM with Adam and Regularization model architecture performance.

No. of Trading Days	Accuracy	Precision	Recall	F1 Score
1000	0.464	0.3973	0.4578	0.3667
1500	0.52	0.5051	0.5018	0.4148
2000	0.504	0.4903	0.4913	0.4825
2500	0.4688	0.2344	0.5	0.3192
3000	0.4747	0.4983	0.4987	0.4522
3500	0.44	0.22	0.5	0.3056
4000	0.467	0.4931	0.4959	0.4227
4500	0.5093	0.5075	0.5076	0.5068
5000	0.5304	0.4925	0.4976	0.4109
entire dataset	0.4871	0.5219	0.5187	0.4787

LSTM architectures. When the dataset size was smaller, the hypertuned model struggled to perform well, but with the increase in dataset size, the model is able to learn the underlying patterns present in the data and give the best accuracy. Its precision, recall, and F1 Score needs improvement.

6.3.4. GRU and Adam

The GRU architecture belongs to the same family of models as the LSTM model. The GRU architecture has shown similar performance to LSTMs and sometimes better performance, depending on the dataset. The next few experiments evaluate the performance of different GRU based architectures. The results of the GRU model optimized with Adam are given in Table 8.

The results in Table 8 show that the GRU optimized with Adam architecture has not performed well. The best accuracy the model got was around 51%. This is low performance when compared to the LSTM based architectures' performance. Regularization needs to be applied to the above model to try and improve its performance.

Table 7
PSO with LSTM and Adam model architecture performance.

No. of Trading Days	Accuracy	Precision	Recall	F1 Score
1000	0.492	0.246	0.5	0.3298
1500	0.4773	0.2387	0.5	0.3231
2000	0.462	0.4759	0.4838	0.4261
2500	0.4688	0.2344	0.5	0.3192
3000	0.5627	0.5515	0.5415	0.5277
3500	0.56	0.28	0.5	0.359
4000	0.455	0.4773	0.4991	0.3204
4500	0.4782	0.492	0.4924	0.4754
5000	0.5264	0.4792	0.4933	0.4043
entire dataset	0.5764	0.2882	0.5	0.3656

Table 8
GRU and Adam model architecture performance.

No. of Trading Days	Accuracy	Precision	Recall	F1 Score
1000	0.48	0.4765	0.4847	0.433
1500	0.5013	0.5074	0.507	0.4962
2000	0.52	0.5128	0.5123	0.5104
2500	0.5152	0.5146	0.5146	0.5144
3000	0.44	0.4496	0.4534	0.4348
3500	0.5166	0.5056	0.5055	0.505
4000	0.511	0.505	0.5049	0.5047
4500	0.5031	0.5061	0.5062	0.5027
5000	0.4864	0.4856	0.4855	0.4851
entire dataset	0.4909	0.5011	0.5011	0.4909

Table 9
GRU with Adam and Regularization model architecture performance.

No. of Trading Days	Accuracy	Precision	Recall	F1 Score
1000	0.508	0.254	0.5	0.3369
1500	0.4773	0.2387	0.5	0.3231
2000	0.462	0.231	0.5	0.316
2500	0.5088	0.5136	0.5134	0.5079
3000	0.4573	0.4914	0.4974	0.3716
3500	0.5589	0.4799	0.4995	0.3632
4000	0.536	0.4433	0.4936	0.3676
4500	0.4898	0.4751	0.4762	0.4736
5000	0.54	0.27	0.5	0.3506
entire dataset	0.5053	0.5146	0.5148	0.5052

6.3.5. GRU with Adam and regularization

This architecture includes the application of regularization to improve generalizability. The GRU architecture is modified by adding dropout and batch normalization layers. These layers will in theory improve the performance of the GRU model. The results of the experiment are given in Table 9.

As observed in Table 9, the results of the experiment show that the regularized GRU architecture is able to perform better than the normal GRU architecture. To further improve performance, hypertuning of the model needs to be explored. The hypertuning of the LSTM architecture significantly improved performance and the same needs to be applied to the GRU architecture.

6.3.6. PSO with GRU and adam

The Particle Swarm Optimization algorithm is used to optimize the GRU architecture. At each hidden layer of the GRU architecture, the algorithm aids in determining the ideal number of GRU units. The ideal number of GRU units was determined using a five particle PSO, and the hypertuned model was built using the results. The hypertuned models' outcomes are listed in Table 10.

As shown in Fig. 7 and Tables 8–10, it can be seen that the hypertuning of the GRU helps in giving good performance, but when comparing its performance to the other GRU architectures, it is observed that the hypertuned GRU is not significantly better than the other GRU architectures.

Table 10
PSO with GRU and Adam model architecture performance.

No. of Trading Days	Accuracy	Precision	Recall	F1 Score
1000	0.508	0.254	0.5	0.3369
1500	0.4773	0.2387	0.5	0.3231
2000	0.502	0.4783	0.4834	0.459
2500	0.5328	0.766	0.5017	0.3507
3000	0.4947	0.5374	0.5239	0.4596
3500	0.4857	0.4859	0.4857	0.484
4000	0.455	0.2275	0.5	0.3127
4500	0.5173	0.4925	0.4938	0.4804
5000	0.5224	0.5134	0.5127	0.5095
entire dataset	0.5257	0.4671	0.4788	0.4454

Table 11
LSTM with GRU ensemble model architecture performance.

No. of Trading Days	Accuracy	Precision	Recall	F1 Score
1000	0.488	0.4116	0.4958	0.3347
1500	0.4747	0.238	0.4972	0.3219
2000	0.542	0.7701	0.5043	0.3593
2500	0.504	0.4817	0.487	0.4541
3000	0.544	0.4841	0.4984	0.3726
3500	0.4994	0.4726	0.4763	0.466
4000	0.445	0.451	0.4788	0.3758
4500	0.4791	0.4846	0.4846	0.4791
5000	0.5432	0.5325	0.5276	0.5159
entire dataset	0.5666	0.4734	0.496	0.3902

Table 12
PSO with LSTM and GRU ensemble model architecture performance.

No. of Trading Days	Accuracy	Precision	Recall	F1 Score
1000	0.492	0.246	0.5	0.3298
1500	0.5227	0.2613	0.5	0.3433
2000	0.47	0.5659	0.5062	0.3442
2500	0.5088	0.5127	0.5126	0.5084
3000	0.4733	0.4824	0.483	0.4718
3500	0.528	0.5139	0.5132	0.5112
4000	0.466	0.4878	0.4918	0.4328
4500	0.4667	0.49	0.492	0.4501
5000	0.4968	0.517	0.514	0.4812
entire dataset	0.4297	0.4683	0.4844	0.3795

6.3.7. LSTM with GRU ensemble

To explore the hypothesis that an ensemble approach is better than a normal approach, the LSTM and GRU architectures were combined with the help of an Average layer that combines the predictions of the model. The performance of the ensemble is given in Table 11.

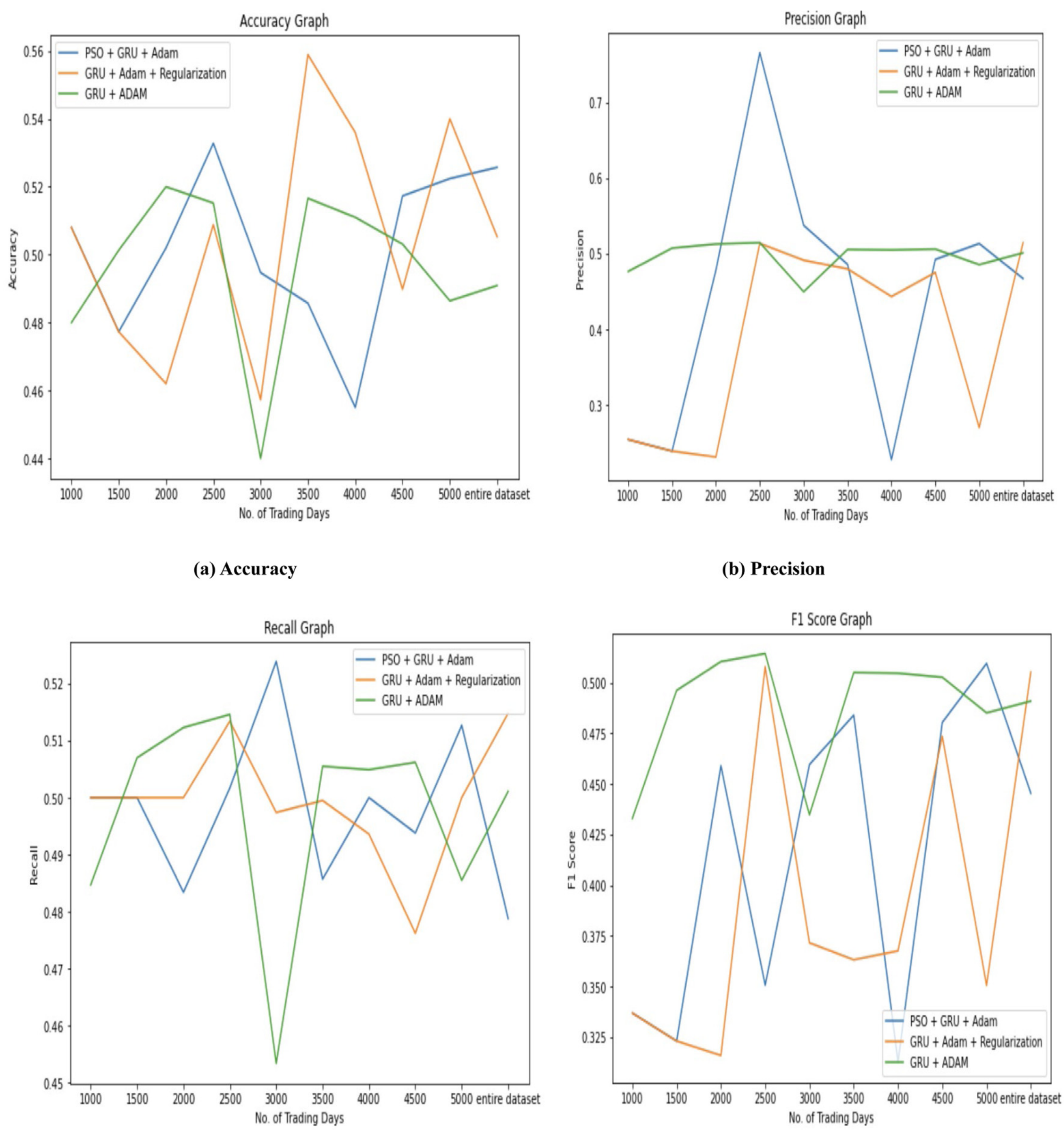
Table 11 shows that the ensemble approach performs well on datasets of different sizes but still has scope for improvement. The above ensemble did not use the hypertuned architecture models for the LSTM and GRU. The next experiment will explore the performance of an ensemble of hypertuned models.

6.3.8. PSO with LSTM and GRU ensemble

This model uses an ensemble of the LSTM and GRU models. The model architectures are modified from the previous experiment by using the output of the PSO algorithm. The hypertuned models are ensembled with the help of averaging. The optimized ensemble model's performance is given below in Table 12.

The results in Tables 11 and 12 show that the ensembling of PSO optimized models did not help create a better model. A normal ensembling of the LSTM and GRU architectures performs significantly better than the PSO optimized ensemble architecture. It is also observed that currently, the performances of PSO + LSTM and LSTM + GRU ensemble architectures show the best results.

In Table 13, it is observed that the PSO + LSTM architecture has a higher accuracy of 0.5764, which shows that the PSO +



(a) Accuracy

(b) Precision

(c) Recall

(d) F1 Score.

Fig. 7. 7(a)–7(d) Performance of different GRU architectures.

Table 13
PSO + LSTM and LSTM + GRU ensemble architecture performance metrics.

Metric	PSO + LSTM	LSTM + GRU ensemble
Accuracy	0.5764	0.5666
Precision	0.2882	0.4734
Recall	0.5	0.496
F1-Score	0.3656	0.3902

LSTM was able to predict stock price movement better than the ensemble. But when observing precision, it is observed that the LSTM + GRU ensemble was able to better predict positive price movement than the POS + LSTM architecture. Based on this discovery, an ensemble of both models was trained to check for any performance improvement.

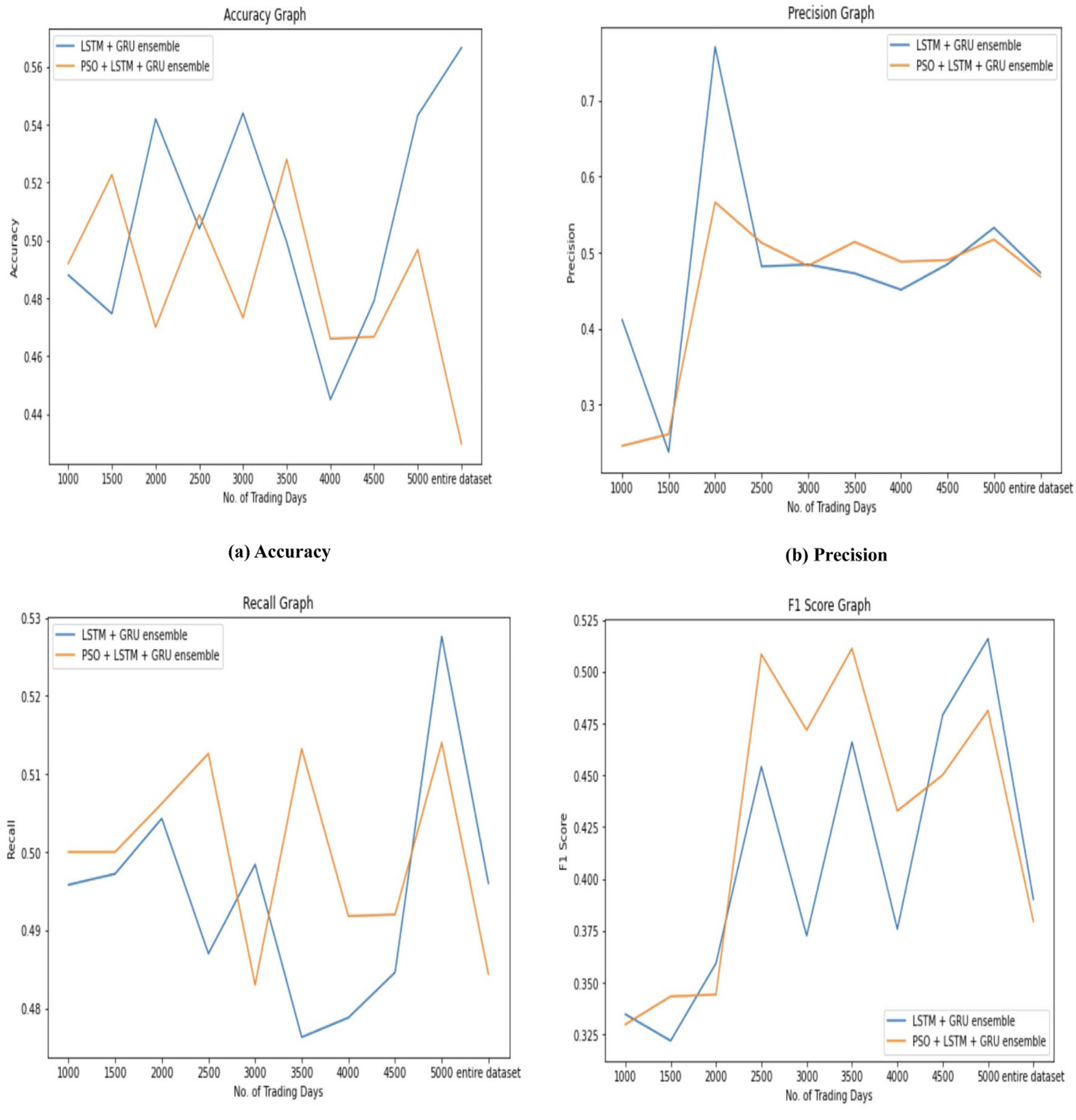
6.3.9. (PSO and LSTM) with (LSTM and GRU) ensemble

In this model architecture, two models that are already an ensemble are assembled. The LSTM + GRU ensemble is combined with the PSO + LSTM model. With the aid of the LSTM + GRU ensemble, the goal of this experiment is to increase the low precision seen in the PSO + LSTM model. Table 14 below provides information about the model's performance.

From Table 15 and Fig. 9, it is observed that the (PSO + LSTM) + (LSTM + GRU) ensemble has given the best accuracy of 0.5772. The model also outperforms PSO + LSTM and LSTM + GRU ensemble with respect to precision, recall, and F1-score. This shows that the ensembling of the two models has been successfully done to create a model that is capable of performing much better than the two base models.

6.4. Results

When examining Fig. 10, it is observed that the (PSO + LSTM) + (LSTM + GRU) ensemble gives the best performance as we



(a) Accuracy **(b) Precision**

(c) Recall **(d) F1 Score**

Fig. 8. 8(a)–8(d) Performance of PSO with LSTM and GRU ensemble model architecture.

Table 14
(PSO + LSTM) + (LSTM + GRU) ensemble architecture performance comparison.

No. of Trading Days	Accuracy	Precision	Recall	F1 Score
1000	0.456	0.4294	0.4515	0.4068
1500	0.5173	0.4603	0.4959	0.3598
2000	0.45	0.4577	0.4723	0.4108
2500	0.512	0.4982	0.4986	0.4805
3000	0.5253	0.5305	0.5305	0.5253
3500	0.5486	0.521	0.5143	0.4866
4000	0.492	0.4843	0.4846	0.4838
4500	0.4756	0.4875	0.488	0.4738
5000	0.5136	0.4915	0.4936	0.4691
entire dataset	0.5772	0.5485	0.531	0.5042

Table 15
PSO + LSTM, LSTM + GRU ensemble, and (PSO + LSTM) + (LSTM + GRU) ensemble architecture performance comparison.

Metric	PSO + LSTM	LSTM + GRU ensemble	(PSO and LSTM) with (LSTM and GRU) ensemble
Accuracy	0.5764	0.5666	0.5772
Precision	0.2882	0.4734	0.5485
Recall	0.5	0.496	0.531
F1-Score	0.3656	0.3902	0.5042

increase the number of trading days and achieves the highest accuracy of 57.72%. Similar results are seen for precision and recall, as depicted in Figs. 11 and 12. This helps in concluding that the (PSO + LSTM) + (LSTM + GRU) ensemble gives the best performance for the task of stock price movement prediction and improves on the performance of the PSO + LSTM and LSTM + GRU ensemble models (see Fig. 13).

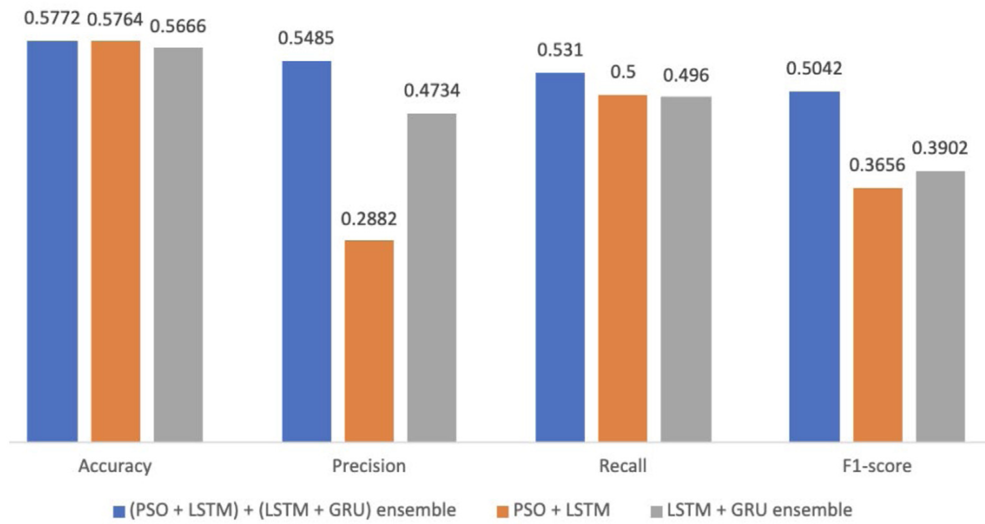


Fig. 9. Performance of (PSO and LSTM) with (LSTM + GRU) ensemble, PSO and LSTM, LSTM with GRU ensemble.

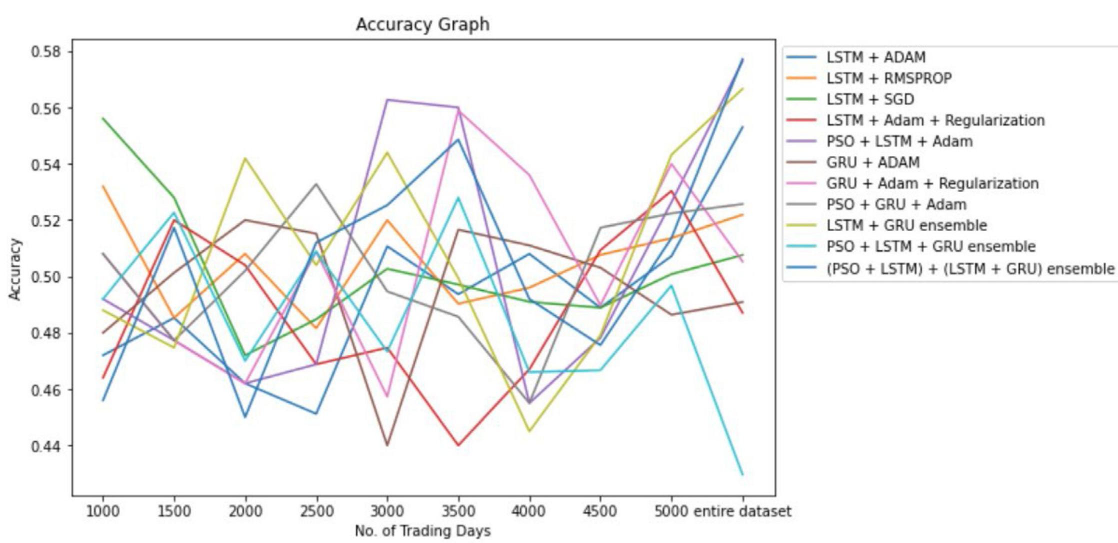


Fig. 10. Accuracy of all architectures.

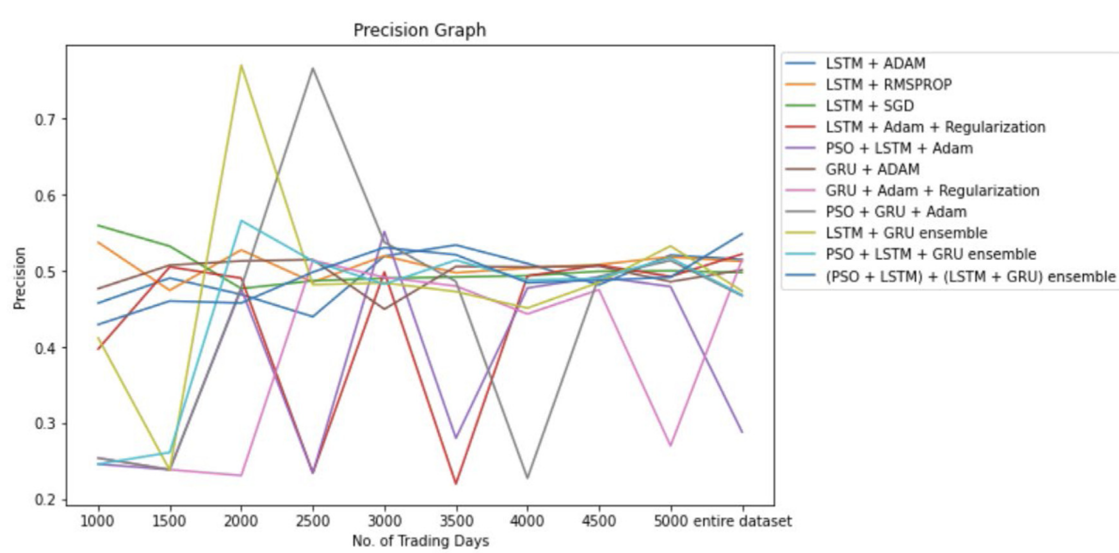


Fig. 11. Precision of all architectures.

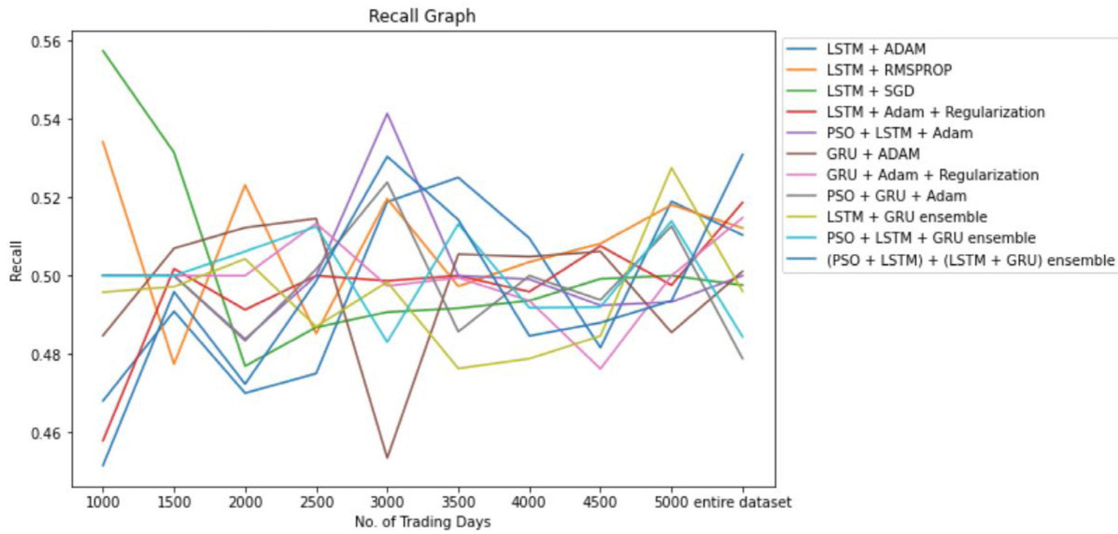


Fig. 12. Recall of all architectures.

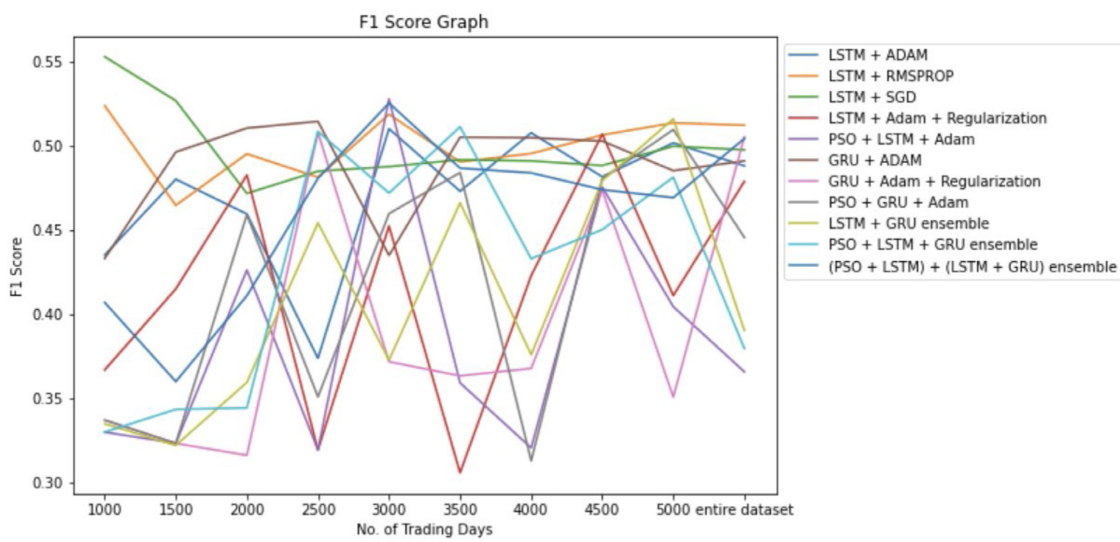


Fig. 13. F1 Score of all architectures.

6.5. Performance comparison

To enhance the critical evaluation of our model, this section compares our proposed models with LSTM based models [28–31] and other architectures. The evaluation aims to examine how our models perform when compared to the methodologies adopted by other researchers. Accuracy, a shared metric among all the studied works, serves as the basis for these comparisons. The comparisons given below in Table 16 examine how the accuracy of our models compares to the accuracy of other proposed works (see Fig. 14).

On comparing our models with other state of the art models that have been trained on similar datasets in Table 16, we can see that the proposed PSO + LSTM model and the (PSO + LSTM) + (LSTM + GRU) ensemble model have performed better than the other models. This shows the capability of using the PSO algorithm to create LSTM models that are capable of predicting stock price movement. Our proposed feature set has been able to show better performance, demonstrating the capability of our approach. The performance of the PSO models and PSO ensembled models is promising, and further research should be done on how the models perform in other deep learning problems.

Table 16
Comparing performance to other researchers' models.

Model	Accuracy
(PSO + LSTM) + (LSTM + GRU) Ensemble	0.5772
PSO + LSTM	0.5764
LSTM + GRU Ensemble	0.5666
ELSTM [11]	0.5720
ALSTM [11]	0.5690
ANN+DT Ensemble [15]	0.5242
WNN [16]	0.5449
CNN [18]	0.5035

6.6. Sample predictions

The predictions produced by the (PSO + LSTM) + (LSTM + GRU) ensemble model are tabulated into two separate tables. One stores the upward stock movement predictions, and the other stores the downward stock movement predictions. Samples of the same are given below in Tables 17 and 18.

The forecasts in Tables 17 and 18 provide a foundation for how the suggested approach might be applied to forecast changes in stock price. The model for a given trading day t, observes

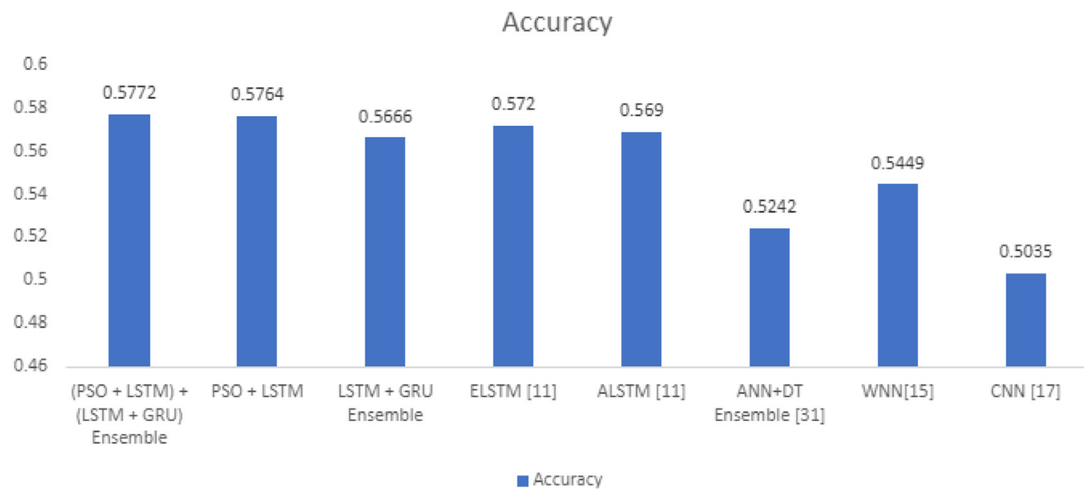


Fig. 14. Comparing performance to other researchers' models.

Table 17
Sample predictions of upward stock price movement.

Close	Volume	Turnover	20_EMA	50_EMA	200_EMA	ITC	HUL	SBI	INFY	HDFC	Diff	Predictions
1454.3	41444271	21873600000.0	1439.1506	1429.6872	1518.1525	720.75	2823.65	227.05	8157.5	251.9	2.65	upward stock price movement
1492.35	41360755	22565400000.0	1446.8754	1433.6635	1517.4214	798.85	2889.5	230.05	8390.5	257.6	22.35	upward stock price movement
1471.45	39155406	20119500000.0	1449.2158	1435.1453	1516.9639	791.8	2837.7	227.4	8310.4	252.35	−20.9	upward stock price movement
1495.25	39577037	25702800000.0	1453.6001	1437.5024	1516.7479	817.15	2704.45	230.65	8583.55	268.65	23.8	upward stock price movement
1511.3	37921171	22623900000.0	1459.0953	1440.3964	1516.6937	809.65	2771.7	233.2	8749.85	278.85	16.05	upward stock price movement
1526.05	51276438	25101800000.0	1465.4719	1443.7553	1516.7868	798.4	286.7	232.6	8600.45	278.0	14.75	upward stock price movement

Table 18
Sample predictions of downward stock price movement.

Close	Volume	Turnover	20_EMA	50_EMA	200_EMA	ITC	HUL	SBI	INFY	HDFC	Diff	Predictions
1517.6	40341969	180154000000.0	1474.8593	1449.4034	1516.795	832.95	276.9	231.95	8487.65	281.8	0.8	downward stock price movement
1509.65	40338082	177442000000.0	1478.1727	1451.766	1516.7239	837.95	279.0	228.7	8296.6	275.7	−7.95	downward stock price movement
1479.65	39534845	226255000000.0	1490.9668	1462.6776	1516.5216	781.7	278.4	224.0	7692.6	254.35	−30.1	downward stock price movement
1463.1	39629219	240476000000.0	1488.3128	1462.6942	1515.9901	805.95	275.95	230.4	7454.55	250.25	−16.55	downward stock price movement
1397.25	47900266	250576000000.0	1470.0217	1457.7038	1513.1222	753.75	256.5	218.0	7349.3	241.6	−26.95	downward stock price movement
1381.25	50255008	273787000000.0	1365.0101	1389.1886	1476.8114	760.3	241.35	198.55	8155.85	238.8	−5.7	downward stock price movement

the above features for a 20 day period and predicts the stock price movement for the next day t+1. These predictions have been separated into different tables, which can then be used by different programs to make trading decisions like going long or going short.

7. Conclusion and future works

In this paper, we proposed a unique approach to stock price movement prediction. The approach involved using features that took into consideration technical information about stocks as well as prices of companies that have large market capitalization in India. Results from this paper show the potential of our proposed feature set and the potential of hypertuning models using metaheuristic algorithms like Particle Swarm Optimization. The (PSO + LSTM) + (LSTM + GRU) ensemble model proposed in this paper provided the best accuracy of 57.72%, precision of 0.5485, recall of 0.531, and F1-Score of 0.5042 when compared to other state of the art methods. PSO hyper parameter optimized models combined with ensembeling performed well for the task of price movement prediction and have the potential to perform well when this approach is be applied to other forms of AI.

Future work will be done on two separate phases of exploration. The first phase will focus on achieving higher levels of accuracy in the task of stock price movement prediction. This will involve examining the use of sentiment analysis of news, as well as including technical indicators used by traders like Bollinger Bands and MACD indicators. Additionally, there will be an exploration into LSTM models with attention layers, along with the use of advanced time series models like N-BEATS [31]. The second phase of work will explore the performance of PSO hyperparameter optimized models, along with ensembling, in different time series related tasks such as speech recognition, weather forecasting, and ECG analysis.

CRediT authorship contribution statement

Akshat Chauhan: Conceptualization, Analysis, Methodology, Investigation, Software, Writing, Reviewing. **Shivaprakash S.J.:** Conceptualization, Analysis, Methodology, Investigation, Software, Writing, Reviewing. **Sabireen H.:** Conceptualization, Analysis, Methodology, Investigation, Visualization, Writing, Reviewing. **Abdul Quadir Md.:** Analysis, Methodology, Resource gathering, Visualization, Writing, Reviewing. **Neelanarayanan Venkataraman:** Resource gathering, Visualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

[1] T.T.L. Chong, W.K. Ng, Technical analysis and the London stock exchange: testing the MACD and RSI rules using the FT30, Appl. Econ. Lett. 15 (14) (2008) 1111–1114.
[2] J. Bollinger, Using bollinger bands, Stocks Commod. 10 (2) (1992) 47–51.
[3] M.J. Horton, Stars, crows, and doji: The use of candlesticks in stock selection, Q. Rev. Econ. Finance 49 (2) (2009) 283–294.
[4] D.M. Nelson, A.C. Pereira, R.A. De Oliveira, Stock market's price movement prediction with LSTM neural networks, in: 2017 International Joint Conference on Neural Networks, IJCNN, Ieee, 2017, pp. 1419–1426.
[5] L.P. Ni, Z.W. Ni, Y.Z. Gao, Stock trend prediction based on fractal feature selection and support vector machine, Expert Syst. Appl. 38 (5) (2011) 5569–5576.
[6] Yang Li, Yi Pan, A novel ensemble deep learning model for stock prediction based on stock prices and news, International Journal of Data Science and Analytics 13 (2) (2022) 139–149.
[7] Wei Li, Ruihan Bao, Keiko Harimoto, Deli Chen, Jingjing Xu, Qi Su, Modeling the stock relation with graph network for overnight stock movement prediction, in: Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence, 2021, pp. 4541–4547.
[8] Thi-Thu Nguyen, Seokhoon Yoon, A novel approach to short-term stock price movement prediction using transfer learning, Appl. Sci. 9 (22) (2019) 4745.
[9] Yi Ji, Alan Wee-Chung Liew, Lixia Yang, A novel improved particle swarm optimization with long-short term memory hybrid model for stock indices forecast, IEEE Access 9 (2021) 23660–23671.
[10] J. Sirignano, R. Cont, Universal features of price formation in financial markets: Perspectives from deep learning, Quant. Finance 19 (9) (2019) 1449–1459.
[11] X. Pang, Y. Zhou, P. Wang, W. Lin, V. Chang, An innovative neural network approach for stock market prediction, J. Supercomput. 76 (2020) 2098–2118.
[12] Pei-Yi Hao, Chien-Feng Kung, Chun-Yang Chang, Jen-Bing Ou, Predicting stock price trends based on financial news articles and using a novel twin support vector machine with fuzzy hyperplane, Appl. Soft Comput. 98 (2021) 106806.
[13] Y. Kara, M.A. Boyacioglu, Ö.K. Baykan, Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the Istanbul stock exchange, Exp. Syst. Appl. 38 (5) (2011) 5311–5319.
[14] Ervandio Irzky Ardyanta, Hasrini SARI, A prediction of stock price movements using support vector machines in Indonesia, J. Asian Finance Econ. Bus. 8 (8) (2021) 399–407.
[15] Chih F. Tsai, Sammy P. Wang, Stock price forecasting by hybrid machine learning techniques, in: Proceedings of the International Multiconference of Engineers and Computer Scientists, Vol. 1, no. 755, 2009.
[16] L. Lei, Wavelet neural network prediction method of stock price trend based on rough set attribute reduction, Appl. Soft Comput. 62 (2018) 923–932.
[17] W. Long, Z. Lu, L. Cui, Deep learning-based feature engineering for stock price movement prediction, Knowl.-Based Syst. 164 (2019) 163–173.
[18] K. Bhardwaj, Convolutional neural network (CNN/ConvNet) in stock price movement prediction, 2021, arXiv preprint arXiv:2106.01920.
[19] A. Alsharef, K. Aggarwal, M. Kumar, A. Mishra, Review of ML and AutoML solutions to forecast time-series data, Arch. Comput. Methods Eng. 29 (7) (2022) 5297–5311.
[20] MR. Hassan, B. Nath, Stock market forecasting using hidden Markov model: A new approach, in: Proceedings—5th International Conference on Intelligent Systems Design and Applications 2005, in: ISDA '05, 2005, pp. 192–196.
[21] S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, in: International Conference on Machine Learning, pmlr, 2015, pp. 448–456.
[22] T. Cooijmans, N. Ballas, C. Laurent, Ç. Gülçehre, A. Courville, Recurrent batch normalization, 2016, arXiv preprint arXiv:1603.09025.
[23] L. Hou, J. Zhu, J. Kwok, F. Gao, T. Qin, T.Y. Liu, Normalization helps training of quantized LSTM, Adv. Neural Inf. Process. Syst. 32 (2019).
[24] Kim Kyoung-jae, Ingoo Han, Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index, Expert Syst. Appl. 19 (2) (2000) 125–132, ISSN: 0957-4174.
[25] R.C. Staudemeyer, E.R. Morris, Understanding LSTM–a tutorial into long short-term memory recurrent neural networks, 2019, arXiv preprint arXiv:1909.09586.
[26] A. Sherstinsky, Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network, Physica D 404 (2020) 132306.
[27] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of ICNN'95-International Conference on Neural Networks,Vol. 4, IEEE, 1995, pp. 1942–1948.
[28] F.A. Gers, J. Schmidhuber, F. Cummins, Learning to forget: Continual prediction with LSTM, Neural Comput. 12 (10) (2000) 2451–2471.
[29] D. Shah, H. Isah, F. Zulkernine, Stock market analysis: A review and taxonomy of prediction techniques, Int. J. Financial Stud. 7 (2019) 26.
[30] Gao Ya, Rong Wang, Enmin Zhou, Stock prediction based on optimized LSTM and GRU models, Sci. Program. 2021 (2021) 4055281, 8.
[31] B.N. Oreshkin, D. Carpov, N. Chapados, Y. Bengio, N-BEATS: Neural basis expansion analysis for interpretable time series forecasting, 2019, arXiv preprint arXiv:1905.10437.